KEVIN CARLOS TENAZOA CUBA

SV73354567

Descripción breve

Este informe explora los conceptos fundamentales de las bases de datos relacionales, relevantes para el curso de programación. Cubrimos las reglas de Codd, la normalización de bases de datos y el modelo relacional, resaltando su importancia en el diseño y gestión eficiente de bases de datos para aplicaciones informáticas.

bases de datos Relacionales

**HACKATON 08 – DESARROLLADOR BACKEND**



HACKATON 08 – DESARROLLADOR BACKEND

**Tabla de Contenido**

[**Introducción 2**](#_Toc166449867)

[**12 Reglas de Codd 3**](#_Toc166449868)

[**Normalización de Base de Datos 5**](#_Toc166449869)

[**El Modelo Relacional para el Modelado y la Gestión de Bases de Datos 5**](#_Toc166449870)

[**Conclusión 6**](#_Toc166449871)

[**Línea De Tiempo 7**](#_Toc166449872)

# **Introducción**

En la era digital actual, las bases de datos relacionales se han convertido en un pilar fundamental en el ámbito de la programación y la gestión de datos. Desde la década de 1970, cuando el matemático Edgar F. Codd introdujo sus innovadoras ideas sobre la organización de datos, estas bases de datos han revolucionado la forma en que almacenamos, organizamos y accedemos a la información. Codd propuso un conjunto de principios, conocidos como las 12 reglas de Codd, que sentaron las bases para la creación de sistemas de gestión de bases de datos relacionales que garantizan la integridad, la consistencia y la eficiencia de los datos.

En este informe, nos sumergiremos en el fascinante mundo de las bases de datos relacionales, explorando no solo las reglas de Codd, sino también la vital práctica de la normalización de bases de datos. La normalización es un proceso esencial que busca eliminar la redundancia y las anomalías en la estructura de la base de datos, asegurando que los datos estén organizados de manera eficiente y coherente. A través de este proceso, se logra una base de datos más robusta y fácil de mantener, lo que facilita tanto la inserción como la recuperación de información.

Además, examinaremos en detalle el modelo relacional, que sirve como el marco conceptual para el diseño y la gestión de bases de datos relacionales. Este modelo, basado en la teoría de conjuntos y la lógica matemática, organiza los datos en tablas que representan entidades del mundo real y las relaciones entre ellas. Al adoptar este enfoque, los programadores pueden crear sistemas de bases de datos flexibles y escalables que se adaptan a una amplia gama de necesidades y aplicaciones.

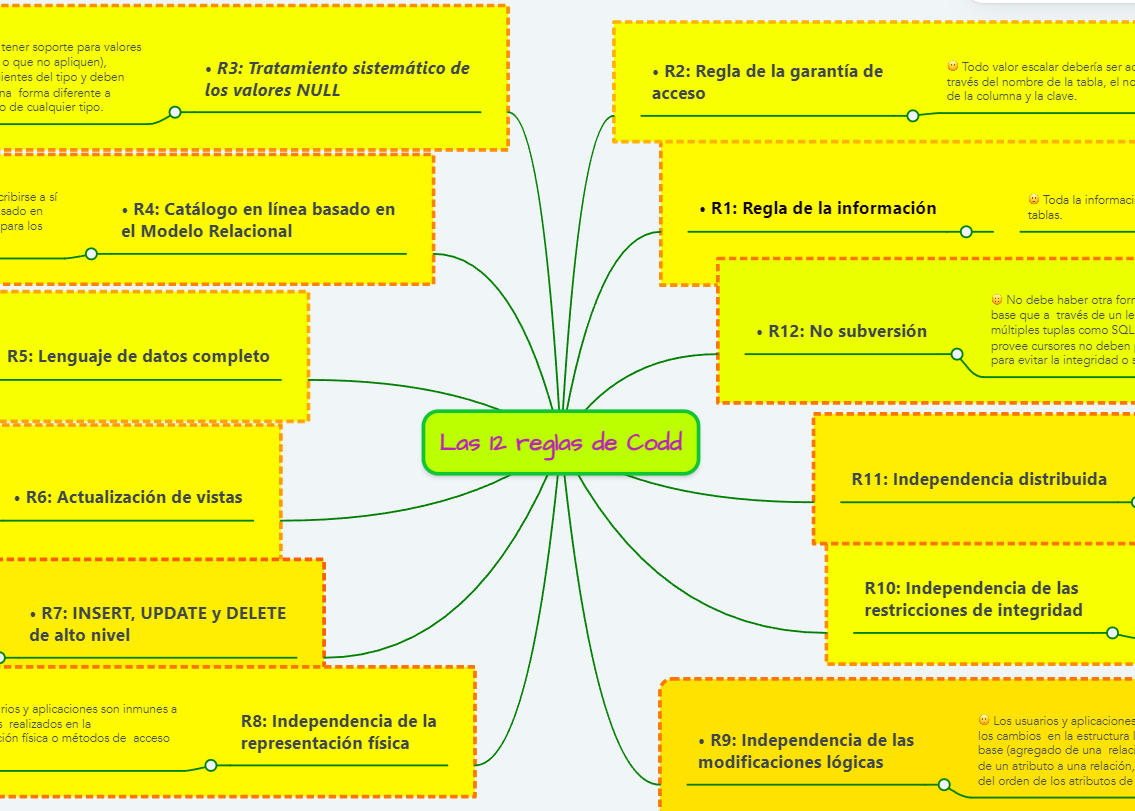
En resumen, este informe ofrece una exploración exhaustiva de los conceptos fundamentales detrás de las bases de datos relacionales, desde las reglas pioneras de Codd hasta las prácticas modernas de normalización y modelado relacional. Al comprender estos principios, los estudiantes y profesionales de la programación pueden desarrollar habilidades sólidas en el diseño y gestión de sistemas de bases de datos, allanando el camino hacia soluciones informáticas más eficientes y robustas.

# **12 Reglas de Codd**

Las 12 reglas de Codd representan un hito fundamental en la teoría de las bases de datos relacionales, estableciendo los criterios esenciales que un sistema de gestión de bases de datos debe cumplir para ser considerado verdaderamente relacional. Estas reglas, propuestas por Edgar F. Codd en la década de 1970, se han convertido en el estándar por el cual se evalúan y se diseñan los sistemas de bases de datos relacionales.

1. **Regla de la Información:** Toda la información en una base de datos relacional debe estar representada de manera explícita en forma de valores en las tablas.
2. **Regla del Acceso Garantizado:** Cada dato debe ser accesible mediante una combinación de nombre de tabla, valor de clave primaria y nombre de columna.
3. **Regla del Sistema Integral:** La integridad del sistema debe mantenerse mediante reglas de integridad definidas en el diccionario de datos.
4. **Regla de la Actualización Integral:** Todas las vistas que son teóricamente actualizables deben ser actualizables por el sistema.
5. **Regla de la Sublenguaje de Datos Completo:** El sistema debe soportar un lenguaje de datos completo que sea capaz de manejar todas las operaciones de consulta y manipulación de datos.
6. **Regla de la Vista Actualizable:** Todas las vistas actualizables deben ser actualizables.
7. **Regla de la Distribución Online:** Las aplicaciones deben funcionar correctamente independientemente de la distribución de los datos.
8. **Regla de la Independencia Física:** Los programas de aplicación deben ser independientes de los cambios en la estructura física de la base de datos.
9. **Regla de la Independencia Lógica:** Los programas de aplicación deben ser independientes de los cambios en la estructura lógica de la base de datos.
10. **Regla de la Independencia de la Integridad:** La integridad de la base de datos debe ser independiente de las aplicaciones que acceden a ella.
11. **Regla de la Independencia de la Distribución:** La distribución de los datos debe ser invisible para los usuarios y las aplicaciones.
12. **Regla de la No Subversión:** Si el sistema proporciona un lenguaje de bajo nivel que permite el acceso a los datos, este lenguaje no puede utilizarse para eludir las reglas de integridad y seguridad del sistema.

Estas reglas son esenciales para garantizar que una base de datos relacional sea coherente, fácil de mantener y proporcione un acceso eficiente a la información. Al seguir estas reglas, los sistemas de gestión de bases de datos pueden ofrecer un alto nivel de confiabilidad y eficacia en la gestión de datos.



# **Normalización de Base de Datos**

La normalización de bases de datos es un proceso fundamental en el diseño y la gestión de bases de datos relacionales, cuyo objetivo principal es eliminar la redundancia de datos y evitar anomalías en la actualización, inserción y eliminación de datos. Al organizar los datos de manera eficiente, la normalización ayuda a garantizar la integridad de la información y facilita el mantenimiento de la base de datos a lo largo del tiempo.

Este proceso se lleva a cabo siguiendo una serie de formas normales, que van desde la primera hasta la quinta forma normal. Cada forma normal define un conjunto específico de reglas y requisitos que deben cumplirse para que una base de datos esté correctamente normalizada.

* **Primera Forma Normal (1FN):** En esta forma normal, cada tabla de la base de datos debe tener una clave primaria única y cada columna debe contener un solo valor atómico, es decir, no debe haber valores repetidos ni múltiples valores en una sola celda.
* **Segunda Forma Normal (2FN**): La segunda forma normal se alcanza cuando una tabla está en 1FN y todos los atributos que no son claves dependen completamente de la clave primaria.
* **Tercera Forma Normal (3FN):** En la tercera forma normal, una tabla debe estar en 2FN y no debe haber dependencias transitivas entre los atributos no clave.
* **Cuarta y Quinta Forma Normal (4FN y 5FN):** Estas formas normales son menos comunes y se centran en situaciones específicas de diseño de bases de datos para eliminar redundancias y mejorar la eficiencia.

La aplicación de la normalización garantiza que la base de datos esté bien estructurada y optimizada para consultas y operaciones. Además, facilita la escalabilidad del sistema y reduce la posibilidad de errores de datos. Sin embargo, es importante tener en cuenta que la normalización puede implicar cierto grado de desnormalización en casos específicos para optimizar el rendimiento en determinadas consultas. En resumen, la normalización es un proceso esencial que ayuda a mantener la coherencia y la eficiencia en la gestión de bases de datos relacionales.

# **El Modelo Relacional para el Modelado y la Gestión de Bases de Datos**

El modelo relacional, propuesto por Edgar F. Codd en la década de 1970, revolucionó la forma en que se diseñan y gestionan las bases de datos. Este enfoque se basa en la teoría de conjuntos y la lógica matemática, y se ha convertido en el estándar de facto para el modelado y gestión de datos en una amplia gama de aplicaciones y sistemas de información. En el modelo relacional, los datos se organizan en tablas, también conocidas como relaciones, donde cada fila representa una entidad única y cada columna representa un atributo o característica de esa entidad. Por ejemplo, en una base de datos de una tienda en línea, podría haber una tabla para clientes con columnas para el nombre, la dirección y el número de teléfono de cada cliente.

Una de las características clave del modelo relacional es la capacidad de establecer relaciones entre tablas. Esto se logra mediante el uso de claves primarias y claves externas. Una clave primaria es un atributo o conjunto de atributos que identifica de manera única cada fila en una tabla. Por otro lado, una clave externa es un atributo que establece una relación entre dos tablas al hacer referencia a la clave primaria de otra tabla. Por ejemplo, en una base de datos de pedidos en línea, la tabla de pedidos podría tener una clave externa que haga referencia a la clave primaria de la tabla de clientes para asociar cada pedido con un cliente específico.

Esta capacidad para representar relaciones entre entidades es una de las fortalezas del modelo relacional, ya que permite modelar de manera efectiva la complejidad del mundo real. Además, el modelo relacional proporciona una estructura flexible y eficiente para almacenar y recuperar datos. Las consultas se realizan mediante un lenguaje de consulta estructurado, como SQL (Structured Query Language), que permite realizar operaciones complejas de búsqueda, inserción, actualización y eliminación de datos de manera eficiente.

# **Conclusión**

Las bases de datos relacionales son una herramienta poderosa para la gestión de datos en sistemas de información. Entender los principios fundamentales detrás de las 12 reglas de Codd, la normalización de bases de datos y el modelo relacional es esencial para diseñar bases de datos eficientes y robustas. Al aplicar estos conceptos, los programadores pueden crear sistemas de información que sean confiables, escalables y fáciles de mantener.

# **Línea De Tiempo**

