Normalización

Bases de datos I – UTN FRRO

Giuliano Crenna

Título: Generalización y Especialización en Diseño de Sistemas

Introducción

En el diseño de sistemas y la programación orientada a objetos, los conceptos de generalización y especialización son fundamentales para construir sistemas flexibles y mantenibles. Estos conceptos permiten manejar la complejidad mediante la creación de jerarquías de clases que facilitan la reutilización del código y la extensión de funcionalidades.

Generalización

La generalización es el proceso mediante el cual se crea una clase general a partir de clases más específicas. Esta técnica se utiliza para reducir la redundancia al permitir que las clases derivadas compartan atributos y métodos comunes. Por ejemplo, en un sistema de vehículos, una clase general llamada Vehículo podría tener atributos como marca, modelo y año, junto con métodos como ***conducir()*** y ***detener().***

Especialización

Por otro lado, la especialización consiste en crear clases más específicas a partir de una clase general. Esto permite añadir o modificar atributos y métodos para cumplir con requisitos particulares. Siguiendo el ejemplo anterior, podríamos tener dos clases especializadas: Automóvil y Bicicleta. La clase Automóvil podría tener un atributo adicional como ***tipo_combustible*** y un método específico como ***abrir_trunk()***, mientras que la clase Bicicleta podría incluir atributos como ***tipo_freno*** y métodos como ***usar_campana()***.

Beneficios de la Generalización y Especialización

1. **Reducción de Código Repetitivo:** Al generalizar, se evita duplicar código que es común a múltiples clases. Esto hace que el mantenimiento del sistema sea más sencillo.
2. **Mejora de la Mantenibilidad:** Los cambios en la clase general se propagan a las clases especializadas, facilitando la actualización y corrección de errores.
3. **Facilitación de la Extensión:** Las nuevas funcionalidades se pueden agregar mediante la especialización, sin necesidad de modificar las clases existentes.

Consideraciones

Es crucial encontrar el equilibrio adecuado entre generalización y especialización. La generalización excesiva puede llevar a una complejidad innecesaria, mientras que la especialización excesiva puede resultar en un sistema difícil de manejar. La decisión sobre cuándo generalizar o especializar debe basarse en el contexto del sistema y los requisitos específicos del proyecto.

Normalización y Desnormalización en el Diseño de Bases de Datos

En el diseño de bases de datos, la normalización y la desnormalización son conceptos fundamentales que garantizan la eficiencia, integridad y consistencia de los datos. Este artículo explora en detalle estos conceptos, proporcionando una comprensión profunda de cómo se aplican en el diseño de bases de datos, junto con ejemplos prácticos.

1. Introducción al Diseño de Bases de Datos

El diseño de bases de datos es un proceso crucial que implica organizar los datos de manera que se minimicen las redundancias y se asegure la integridad de los datos. Uno de los enfoques más utilizados para lograr esto es la normalización. Este proceso involucra aplicar reglas (formas normales) para estructurar las tablas de la base de datos de manera eficiente. Sin embargo, en ciertos casos, es necesario desnormalizar las tablas para optimizar el rendimiento de consultas.

¿Qué es la Normalización?

La normalización es un proceso sistemático de organización de los datos en una base de datos para reducir la redundancia y mejorar la integridad de los datos. Fue introducido por Edgar F. Codd en 1970, quien propuso una serie de reglas conocidas como formas normales para guiar el proceso de normalización.

Objetivos de la Normalización:

- **Eliminar Redundancias:** La normalización ayuda a reducir la duplicidad de datos al dividir tablas grandes en tablas más pequeñas y bien definidas.
- **Mejorar la Consistencia:** Al eliminar redundancias, se evita la posibilidad de inconsistencia en los datos, garantizando que la misma información no se almacene en múltiples lugares.
- **Facilitar el Mantenimiento:** Una base de datos normalizada es más fácil de mantener y actualizar, ya que cualquier cambio en los datos debe realizarse en un solo lugar.
-

Desventajas de la Normalización:

Aunque la normalización mejora la integridad y consistencia, también puede tener desventajas, como:

- **Complejidad en las Consultas:** Las consultas en una base de datos altamente normalizada pueden volverse más complejas, ya que requieren unir múltiples tablas.
- **Impacto en el Rendimiento:** La normalización puede reducir el rendimiento de las consultas en bases de datos de gran tamaño, especialmente en entornos de análisis de datos donde se requiere acceder a grandes volúmenes de información rápidamente.

Formas Normales

Las formas normales son un conjunto de reglas que definen el nivel de normalización en una base de datos. A continuación, se detallan las formas normales más comunes: Primera Forma Normal (1FN), Segunda Forma Normal (2FN), Tercera Forma Normal (3FN), y Forma Normal de Boyce-Codd (BCNF).

Primera Forma Normal (1FN)

La Primera Forma Normal (1FN) establece que una tabla está en 1FN si cumple con los siguientes criterios:

- **Valores Atómicos:** Cada columna debe contener valores indivisibles o atómicos, es decir, no se permiten atributos multivaluados (más de un valor en una sola columna).
- **Filas Únicas:** No debe haber filas duplicadas en la tabla.
- **No Grupos Repetidos:** No debe haber conjuntos repetidos de columnas.

nombre	teléfono
John Smith	45 35 45 12 35 46 78 98
Carmen Aguilar	55 25 12 45 54 36 11 28

nombre	teléfono
John Smith	45 35 45 12
John Smith	35 46 78 98
Carmen Aguilar	55 25 12 45
Carmen Aguilar	54 36 11 28

Segunda Forma Normal (2FN)

La Segunda Forma Normal (2FN) extiende la 1FN y agrega la condición de que no debe haber dependencias parciales de la clave primaria. Es decir, todos los atributos no clave deben depender completamente de la clave primaria.

Tercera Forma Normal (3FN)

La Tercera Forma Normal (3FN) establece que una tabla está en 3FN si está en 2FN y no tiene dependencias transitivas. Una dependencia transitiva ocurre cuando un atributo no clave depende de otro atributo no clave.

ALUMNOS MATRICULADOS				
rut	nombre	apellido	cod_curso	descripcion
1-9	Pedro	Pérez	AE600	Algoritmos y Estructuras de datos
2-7	Juan	Jara	BD253	Bases de Datos
2-7	Juan	Jara	AE600	Algoritmos y Estructuras de datos
3-5	Diego	Díaz	BD253	Bases de Datos
4-4	Maria	Martinez	BD253	Bases de Datos

ALUMNO		
rut	nombre	apellido
1-9	Pedro	Pérez
2-7	Juan	Jara
3-5	Diego	Díaz
4-4	Maria	Martinez

CURSO	
cod_curso	descripcion
AE600	Algoritmos y Estructuras de datos
BD253	Bases de Datos

MATRICULA	
rut	cod_curso
1-9	AE600
2-7	BD253
2-7	AE600
3-5	BD253
4-4	BD253

Forma Normal de Boyce-Codd (BCNF)

La Forma Normal de Boyce-Codd (BCNF) es una versión más estricta de la 3FN. Una tabla está en BCNF si, para cada dependencia funcional X → Y, X es una superclave. Esto significa que cualquier dependencia funcional debe estar basada en una clave candidata, no solo en la clave primaria.

Normalización vs. Desnormalización

A pesar de los beneficios de la normalización, hay situaciones en las que la desnormalización puede ser preferible. La desnormalización es el proceso de combinar tablas previamente normalizadas para mejorar el rendimiento de las consultas, reduciendo la necesidad de unir múltiples tablas.

Ventajas de la Desnormalización:

- **Mejora del Rendimiento:** En entornos donde el rendimiento de las consultas es crítico, como en bases de datos analíticas, la desnormalización puede reducir el tiempo de respuesta al minimizar el número de uniones.
- **Simplicidad en las Consultas:** Las consultas en bases de datos desnormalizadas suelen ser más sencillas, ya que los datos necesarios se encuentran en menos tablas.

Desventajas de la Desnormalización:

- Mayor Redundancia: La desnormalización introduce redundancia de datos, lo que puede llevar a inconsistencias si no se maneja correctamente.
- Complejidad en el Mantenimiento: Actualizar datos en una base de datos desnormalizada puede ser más complejo, ya que los cambios deben reflejarse en múltiples lugares.

Casos de Uso de la Desnormalización:

La desnormalización es común en bases de datos orientadas a análisis o en situaciones donde la velocidad de las consultas es más importante que la consistencia estricta de los datos. Un ejemplo típico es un almacén de datos donde se almacenan grandes volúmenes de datos históricos para informes y análisis.

Ejemplos Prácticos: Normalización y Desnormalización

Ejercicio de Normalización:

Supongamos una tabla desnormalizada que contiene información de ventas:

ID Venta	Fecha	Cliente	Producto	Cantidad	Precio Unitario	Total
1	01/08/2024	Juan Pérez	Televisor	2	$500	$1000
2	02/08/2024	María López	Lavadora	1	$300	$300

Para normalizar esta tabla hasta la 3FN, seguiríamos los siguientes pasos:

1. **1FN:** Asegurarnos de que todos los atributos contienen valores atómicos (lo cumple).
2. **2FN:** Eliminar dependencias parciales (dividir en tablas de ventas y productos).
3. **3FN:** Eliminar dependencias transitivas (crear una tabla separada para clientes).

Bibliografía:

- **"Database System Concepts"** - *Abraham Silberschatz, Henry F. Korth, S. Sudarshan*
- **"Fundamentals of Database Systems"** - *Ramez Elmasri, Shamkant B. Navathe*
- **"Database Design for Mere Mortals"** - *Michael J. Hernandez*