FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

INVESTIGACIÓN: NORMALIZACIÓN

Estudiantes del curso ITIZ2200
Base de Datos
Universidad de Las Américas, Quito-Ecuador

Daniel Sierra, Ricardo Herrera y Pablo Vargas

07/05/2025

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

I. ÍNDICE

II. INTRODUCCIÓN

La normalización de bases de datos es un proceso fundamental en el diseño de sistemas de información, cuyo objetivo principal es organizar y estructurar eficientemente los datos para minimizar la redundancia y evitar anomalías en las operaciones de inserción, actualización y eliminación. Este proceso se basa en un conjunto de reglas llamadas formas normales, que permiten descomponer las tablas en estructuras más simples y consistentes, facilitando así la integridad y consistencia de la información almacenada.

El concepto de normalización fue introducido por Edgar F. Codd, creador del modelo relacional, como una técnica para mejorar la calidad del diseño lógico de bases de datos. Mediante la aplicación de las diversas formas normales, se busca eliminar las dependencias no deseadas entre los datos, contribuyendo a un modelo más robusto y flexible. En este contexto, las dependencias funcionales desempeñan un papel crucial, ya que permiten identificar relaciones entre atributos y determinar la estructura óptima de la tabla.

Este trabajo tiene como objetivo ayudar a los estudiantes a comprender y aplicar los principios de normalización hasta la Tercera Forma Normal (3NF), identificando redundancias, dependencias funcionales y cómo mejorar el diseño lógico de una base de datos. A través de la investigación y el análisis de ejemplos prácticos, se explorarán los beneficios de una base de datos normalizada y se discutirán brevemente los casos en los que podría ser necesaria la desnormalización.

III. DESARROLLO

Introducción a la normalización

La normalización de bases de datos es un proceso esencial en el diseño de sistemas de información. Su objetivo principal es organizar y estructurar eficientemente los datos para minimizar la redundancia y evitar anomalías en las operaciones de inserción, actualización y eliminación. Este proceso se basa en un conjunto de reglas llamadas formas normales, que permiten descomponer las tablas en estructuras más simples y consistentes, facilitando así la integridad y consistencia de la información almacenada.

El concepto de normalización fue introducido por Edgar F. Codd, creador del modelo relacional, como una técnica para mejorar la calidad del diseño lógico de bases de datos. Mediante la aplicación de las diversas formas normales, buscamos eliminar dependencias no deseadas entre los datos, contribuyendo a un modelo más robusto y flexible. En este contexto, las dependencias funcionales desempeñan un papel crucial, ya que permiten identificar relaciones entre atributos y determinar la estructura óptima de la tabla.

Este trabajo tiene como objetivo ayudar a los estudiantes a comprender y

aplicar los principios de normalización hasta la Tercera Forma Normal (3NF), identificando redundancias, dependencias funcionales y cómo mejorar el diseño lógico de una base de datos. A través de la investigación y el análisis de ejemplos prácticos, se explorarán los beneficios de una base de datos normalizada y se discutirán brevemente los casos en los que podría ser necesaria la desnormalización.

Formas normales

Las formas normales son principios fundamentales en el diseño de bases de datos relacionales, que permiten la redundancia para los requisitos y la integración de datos para las tareas. Las tres primeras formas normales se presentan en detalle:

Primera Forma Normal (1FN)
Definición:
Una tabla está en 1FN si:

- Todos los atributos contienen valores atómicos (indivisibles).
- No existen grupos repetitivos.
- Cada registro es único.

Reglas:

- Eliminar campos multivaluados.
- Hay que asegurar que cada campo contenga un solo valor.
- Establecer una clave primaria única para cada fila.

Ejemplo:
Antes de aplicar 1FN:

ID_Empleado	Nombre	Teléfonos
1	Ana	0991234567, 0987654321
2	Luis	0976543210

Después de aplicar 1FN:

ID_Empleado	Nombre	Teléfono
1	Ana	0991234567
1	Ana	0987654321
2	Luis	0976543210

En este ejemplo, se ha descompuesto el campo multivaluado "Teléfonos" en registros individuales, asegurando que cada campo contenga un solo valor atómico.

Segunda Forma Normal (2FN)
Definición:
Una tabla está en 2FN si:

- Está en 1FN.
- Todos los atributos no clave dependen completamente de la clave primaria.

Reglas:

- Eliminar dependencias parciales.
- Separar los datos en tablas adicionales si es necesario.

Ejemplo:
Antes de aplicar 2FN:

ID_Empleado	ID_Proyecto	Nombre_Empleado	Horas_Trabajadas
1	101	Ana	40
2	102	Luis	35

En este caso, "Nombre_Empleado" depende solo de "ID_Empleado", no de la combinación de "ID_Empleado" y "ID_Proyecto".

Después de aplicar 2FN:
Tabla Empleados:

ID_Empleado	Nombre_Empleado
1	Ana
2	Luis

Tabla Proyectos:

ID_Empleado	ID_Proyecto	Horas_Trabajadas
1	101	40
2	102	35

Se han separado los datos en dos tablas para eliminar la dependencia parcial.

Tercera Forma Normal (3FN)
Definición:
Una tabla está en 3FN

- Está en 2FN.
- No existen dependencias transitivas entre atributos no clave.

Reglas:

- Eliminar dependencias transitivas.
- Crear tablas adicionales para atributos que dependan de otros atributos no clave.

Ejemplo:
Antes de aplicar 3FN:

ID_Empleado	Nombre	ID_Departamento	Nombre_Departamento
1	Ana	10	Recursos Humanos
2	Luis	20	Finanzas

Aquí, "Nombre_Departamento" depende de "ID_Departamento", que a su vez depende de "ID_Empleado".

Después de aplicar 3FN:

Tabla Empleados:

ID_Empleado	Nombre	ID_Departamento
1	Ana	10
2	Luis	20

Tabla Departamentos:

ID_Departamento	Nombre_Departamento
10	Recursos Humanos
20	Finanzas

Se ha eliminado la dependencia transitiva al crear una tabla separada para los departamentos.

Dependencias funcionales

Antes de profundizar en el tema, cabe recordar que la dependencia funcional es la piedra angular de la normalización: es el mecanismo que permite detectar cuándo un atributo (o conjunto de atributos) determina de manera única a otro y, por tanto, cuándo existen posibles redundancias o anomalías en una tabla. Para aplicar correctamente 2FN y 3FN, es esencial identificar y describir todos los FD de cada relación.

Una dependencia funcional es una restricción lógica entre dos conjuntos de atributos de una misma relación, que indica que el valor de un conjunto (el determinante) determina de manera única el valor de otro conjunto (el dependiente). Matemáticamente, para una relación R y sus subconjuntos X, Y ⊆ R, X → Y, decimos que X → Y se da si, para cualquier par de tuplas en R, compartir los mismos valores en X implica compartir los mismos valores. En la práctica, si conocemos el valor de X, podemos buscar en la tabla cualquier tupla correspondiente en X y obtener sin ambigüedad el valor de Y. Por ejemplo, en una tabla Empleados, el número de empleado (EmpleadoID) determina el nombre (EmpleadoName):

Por ejemplo:
EmpleadoID → EmpleadoName
lo que garantiza que no existan dos filas con mismo EmpleadoID pero distinto EmpleadoName

¿Como se representan?

La notación más común para una DF es la flecha "→" que se lee "X determina Y" o "Y" depende funcionalmente de "X". Cuando el determinante está formado por varios atributos, se agrupan a la izquierda de la flecha;

(EmpleadoID, OficinaID) → Empresa

indica que la combinación EmpleadoID + OficinaID determina la empresa. Esta forma de representación permite listar todas las DFs de una relación como un conjunto:

F = { A → B, C D → E, … }

Importancia de aplicar 2FN y 3FN

- **2FN: eliminación de dependencias parciales**

La segunda forma normal requiere que una relación no tenga dependencias funcionales parciales, es decir, ningún atributo que no sea clave depende de ninguna parte (apropiada) de una clave primaria compuesta. Para detectarlos, enumeramos los DF y verificamos si existe un X → Y donde X es un subconjunto adecuado de la clave principal.

Si identificamos K1 K2 → A pero también K1 → A, entonces A depende sólo de K1 (parte

de la clave), lo que viola 2FN.

Eliminamos esta anomalía descomponiendo la relación en dos: una con (K1, A) y otra con (K1, K2, resto)....

Sin la identificación correcta de los DF, no podríamos saber qué atributos extraer para satisfacer el 2FN.

- **3FN: Eliminación de dependencias transitivas**

La tercera forma normal extiende la limpieza de DF al no permitir dependencias transitivas: un atributo no clave no debe depender de otro atributo no clave.

Partimos de DF: si existen X → Y e Y → Z, con X como clave (o superclave) e Y como atributo no clave, entonces Z depende transitivamente de X (X → Z vía Y).

Para corregir esto, separamos Y y Z en una relación independiente.

De nuevo, sin una lista exhaustiva de DF, sería imposible identificar pasajes del tipo "clave → no clave → no clave → no clave" y por tanto llegar a 3FN.

Ventajas y desventajas

Ventajas

- **Eliminación de redundancia de datos:**
  Al dividir datos en múltiples tablas relacionadas, se evita almacenar la misma información. Esto reduce errores y mejora la coherencia.
- **Integridad en los datos:**
  Gracias al uso de herramientas como las Primary y Foreign Keys, se asegura que las relaciones entre datos estén bien definidas, evitando inconsistencias.
- **Facilidad de Mantenimiento:**
  Al tener menor cantidad de datos duplicados esto significa que existe un menor consumo de espacio en el disco especialmente para cuando se está trabajando con bases grandes
- **Flexibilidad y escalabilidad:**
  Al tener una estructura Normalizada nos puede ayudar a la hora de adaptarnos de mejor manera a cambios en los requisitos, ya que esta se basa en un diseño modular.

Desventajas

- **Complejidad en consultas:**
  Al estar divida en muchas tablas, las consultas pueden llegar a ser más complejas además de aumentar su tiempo de respuesta
- **Rendimiento en lectura pueden verse afectado:**
  En sistemas que realizan muchas consultas de lectura, las múltiples uniones entre tablas pueden ralentizar el acceso a los datos.
- Dificultad para usuarios sin **conocimiento:**
  Usuarios sin conocimiento en SQL o modelado relacional pueden tener problemas para entender o manipular

- **Desempeño en entornos distribuidos**
  En bases distribuidas, acceder a las tablas relacionadas en diferentes nodos puede aumentar la latencia y tiempos de espera casos donde la desnormalización puede ser necesaria.
- Aplicaciones de data Warehouse, donde se optimiza lectura rápida

IV. CONCLUSIONES

Tras investigar y trabajar en este proyecto de normalización de bases de datos relacionales, nos dimos cuenta de que este proceso es fundamental para garantizar la integridad y eficacia de la información en cualquier sistema de gestión de bases de datos.
La aplicación de las tres primeras formas normales (1FN, 2FN y 3FN) permite estructurar los datos de forma lógica, eliminando redundancias y evitando problemas comunes como las anomalías de inserción, actualización y eliminación.
Además, aprendimos que las dependencias funcionales son esenciales para identificar las relaciones entre atributos y determinar la estructura óptima de las tablas.
En resumen, la normalización no sólo mejora la calidad del diseño lógico de una base de datos, sino que también facilita su mantenimiento y escalabilidad a largo plazo.
Este trabajo nos ha ayudado a comprender la importancia de aplicar correctamente los principios de normalización para desarrollar sistemas de información más robustos y eficientes.

V. BIBLIOGRAFÍAS

Castañeda, M. P. (s. f.). *Normalización de bases de datos*.

https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/872/mod_resource/content/7/Contenido/index.html

Codd, E. F. (1970). *A relational model of data for large shared data banks*.

Communications of the ACM, *13*(6), 377–387.

https://doi.org/10.1145/362384.362685duitdesign.com+3Software Engineering Stack Exchange+3Wikipedia, la enciclopedia libre+3

Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of database systems* (7ª ed.).

Pearson Education.

Gomez, E. E. P. (2023, 7 agosto). *Normalización de base de datos: formas normales 1nf 2nf 3nf ejemplos de tablas*. freeCodeCamp.org.

https://www.freecodecamp.org/espanol/news/normalizacion-de-base-de-datos-formas-normales-1nf-2nf-3nf-ejemplos-de-tablas/

Helenclu. (s. f.). *Descripción de la normalización de la base de datos - Microsoft 365 Apps*. Microsoft Learn. https://learn.microsoft.com/es-es/office/troubleshoot/access/database-normalization-description

Saavedra, J. A. (2023, 16 julio). *¿Qué es la normalización de bases de datos y cómo hacerla?* Ebac. https://ebac.mx/blog/normalizacion-de-bases-de-datos