|  |  |
| --- | --- |
|  | Trabajo grupal o individual |

## Identificación del trabajo

|  |  |
| --- | --- |
| 1. **Módulo:** | 2 |
| 1. **Asignatura:** | Bases de Datos |
| 1. **RA:** | Crear, popular y relacionar tablas de BD |
| 1. **Docente Online:** | Jacob Felipe |
| 1. **Fecha de entrega**: | 14 de noviembre de 2024 |
|  |  |
|  |  |

## Identificación del/los estudiante/s

|  |  |
| --- | --- |
| **Nombre y apellido** | **Carrera** |
| Christian Aránguiz | Técnico en Programación |

## Introducción

En esta segunda actividad de la asignatura de bases de datos, continuamos el desarrollo de la base de datos **Biblioteca** para la Universidad Puls-AR, proyecto iniciado con la instalación de PostgreSQL mediante Docker-Compose. La primera fase se centró en configurar el entorno y crear la base de datos, permitiéndonos contar con un sistema de almacenamiento eficiente para gestionar los recursos bibliográficos de la universidad.

En esta nueva etapa, nos enfocamos en definir las entidades principales, sus relaciones y las normalizaciones necesarias para asegurar la integridad y eficacia del sistema. Para comenzar, se crearon dos tablas principales: **Lectores** y **Libros**. Sin embargo, este diseño inicial presentó ciertas limitaciones para cumplir con los requerimientos del sistema. Al no existir un registro que relacione directamente a **Lectores** con **Libros**, no sería posible almacenar información detallada sobre los préstamos, como las fechas de préstamo y devolución o el estado del libro. Esta falta de relación impide el seguimiento de los libros prestados a cada lector y la gestión precisa de la disponibilidad de los recursos.

Para resolver esta problemática y garantizar la integridad de los datos, aplicamos la tercera forma normal (3NF), que elimina redundancias y asegura que cada dato dependa de una clave primaria única. En este caso, creamos una tabla intermedia denominada **Préstamos**. Esta tabla permite vincular cada **Lector** con un **Libro** específico, registrando un préstamo único para cada combinación. **Préstamos** no solo conecta **Lectores** y **Libros**, sino que también permite almacenar información adicional de cada transacción, como la fecha de préstamo y devolución.

La tabla **Préstamos** se diseñó bajo las reglas de la tercera forma normal (3NF), eliminando redundancias y asegurando la dependencia funcional entre sus campos, lo que nos permitirá manejar la información de forma óptima y escalable. Este informe detalla el proceso y las decisiones tomadas en la construcción y normalización de esta estructura intermedia, fundamentales para el avance de la base de datos.

## Desarrollo

En esta segunda fase de la construcción de la base de datos biblioteca, el objetivo principal fue definir las entidades principales, sus atributos y las relaciones entre ellas, a fin de estructurar la información de manera lógica y accesible. Para lograr esto, se crearon dos tablas principales en PostgreSQL: libros y lectores. Para conectar estos datos y permitir la gestión de préstamos en el sistema, se diseñó una tabla de relación llamada prestamos, que facilita el seguimiento de qué libros han sido prestados a qué lectores y permite controlar la disponibilidad de los recursos bibliográficos.

1. Creación de la Tabla Libros: La tabla libros ya había sido creada en la primera actividad, donde se definieron los campos básicos para almacenar información relevante sobre cada ejemplar en la biblioteca, como el título, autor, editorial, ISBN y año de publicación (como se había solicitado en la introducción del curso). En esta actividad, se le añadió el campo id\_libro como clave primaria mediante un campo serial autoincremental, lo que garantiza la unicidad de cada registro. Además, se implementó una restricción de clave primaria (PK) en id\_libro, facilitando la referencia de cada libro en otras partes de la base de datos, como en la tabla de Préstamos.

A blue background with white text

Description automatically generated

imagen 1: Imagen de los requisitos en los que se piden el año de publicación del libro

A screen shot of a computer program

Description automatically generated

Imagen 1: Creación de la tabla libros, descripción de la misma, adición de PK y selección la tabla.

1. Creación de la Tabla Lectores: La tabla lectores recoge los datos personales de cada usuario de la biblioteca, como nombre, apellido, correo electrónico y fecha de nacimiento. Al igual que en libros, se implementó un campo id\_lector como clave primaria serial, permitiendo que cada lector tenga un identificador único. Este enfoque facilita la conexión de cada lector con los préstamos que ha realizado, mediante la tabla intermedia.

A screen shot of a computer program

Description automatically generated

imagen 2: Creación de la tabla lectores, descripción de la misma, adición de PK y selección la tabla.

1. Creación de la Tabla Intermedia Préstamos: La tabla prestamos fue diseñada para gestionar la relación de muchos a muchos entre libros y lectores, permitiendo que un lector pueda solicitar múltiples libros y que un libro pueda ser prestado a múltiples lectores. La tabla contiene el campo id\_prestamo como clave primaria y dos claves foráneas, id\_lector y id\_libro, que referencian las tablas lectores y libros, respectivamente. Además, incluye campos para registrar la fecha de préstamo (fecha\_prestamo), la fecha de devolución (fecha\_devolucion), y el estado del préstamo (por ejemplo, si el libro ha sido devuelto o aún está en préstamo).

A screen shot of a computer program

Description automatically generated

1. Normalización y Consistencia: La estructura de la base de datos se diseñó siguiendo la tercera forma normal (3NF), eliminando redundancias y asegurando que cada campo depende únicamente de la clave primaria de la tabla. Esto permite que la información esté organizada y evita inconsistencias. La tabla Préstamos cumple un rol fundamental en este sentido, manteniendo la integridad referencial mediante las claves foráneas y permitiendo el rastreo preciso de los préstamos realizados por los lectores.

A screen shot of a computer screen

Description automatically generated

imagen 3: Diagrama UML mostrando la relación 3NF entre Libros, Lectores y Préstamos..

1. Inserción de Datos Iniciales: Finalmente, se realizaron inserciones de datos iniciales para poblar las tablas Libros y Lectores con un conjunto de registros representativos. Estos datos proporcionan una base para probar las funcionalidades de consulta y relación en la base de datos, permitiendo verificar que el diseño es adecuado para las operaciones necesarias en una biblioteca.

A screenshot of a computer screen

Description automatically generated

## Conclusión

Para finalizar, esta segunda actividad en la creación de la base de datos Biblioteca permitió establecer una estructura sólida y bien organizada para la gestión de los recursos bibliográficos de la Universidad Puls-AR. Tras la configuración inicial del entorno en PostgreSQL, nos enfocamos en la definición y normalización de las entidades clave y sus relaciones. La creación de las tablas Lectores y Libros proporcionó una base esencial, aunque limitada para gestionar los préstamos. Para resolver las restricciones de este diseño inicial, se añadió la tabla Préstamos, la cual actúa como un registro de cada transacción de préstamo y permite almacenar información detallada sobre cada uno de ellos, como las fechas y el estado del préstamo.

La implementación de la tercera forma normal (3NF) en esta estructura ha sido clave para garantizar la integridad y evitar redundancias, permitiendo que los datos se gestionen de manera óptima y escalable. Esta organización no solo facilita el seguimiento de los préstamos y la disponibilidad de los libros, sino que también deja la base lista para futuras ampliaciones en las próximas fases. Así, este diseño asegura que el sistema bibliotecario cumpla con las necesidades operativas de la universidad, proporcionando un modelo de datos eficiente y sostenible.

## Bibliografía

* PostgreSQL Global Development Group. (n.d.). *PostgreSQL Documentation*. Recuperado de <https://www.postgresql.org/docs>
* FreeCodeCamp Normalización de bases de datos: Formas normales 1nf, 2nf, 3nf. <https://www.freecodecamp.org/espanol/news/normalizacion-de-base-de-datos-formas-normales-1nf-2nf-3nf-ejemplos-de-tablas/>