**Base de Datos de Biblioteca**

Integrantes: Pamela Salazar, Emanuel Clementi.

Este documento describe la estructura y funcionalidad de una base de datos diseñada para gestionar los préstamos en una biblioteca. La base de datos permite llevar un registro de libros, usuarios, préstamos y auditorías. Esta base de datos permite:

**Registro y Gestión de Libros**: Almacenar información sobre cada libro, incluyendo título, autor, ISBN, categoría, editorial, fecha de publicación y estado.

**Gestión de Usuarios**: Mantener información sobre los usuarios, como nombre, apellido, dirección, teléfono y correo electrónico.

**Gestión de Préstamos**: Registrar y controlar los préstamos de libros, incluyendo datos del usuario, libro prestado, bibliotecario que realiza el préstamo, fecha de préstamo, fecha de vencimiento y fecha de devolución.

**Auditoría de Préstamos**: Registrar acciones realizadas sobre los préstamos para fines de seguimiento y control.

El sistema se compone de las siguientes entidades:

**Libro**:

Almacena información sobre los libros disponibles en la biblioteca.

**Campos:**

id\_libro: Identificador único del libro (clave primaria).

titulo: Título del libro.

autor: Autor del libro.

isbn: Código ISBN del libro.

categoria: Categoría o género del libro.

editorial: Editorial que publicó el libro.

fecha\_de\_publicacion: Fecha en que el libro fue publicado.

estado: Estado del libro (por ejemplo, disponible, prestado).

**Usuario**

Almacena información sobre los usuarios que pueden tomar prestados libros de la biblioteca.

**Campos:**

id\_usuario: Identificador único del usuario (clave primaria).

nombre: Nombre del usuario.

apellido: Apellido del usuario.

direccion: Dirección del usuario.

telefono: Número de teléfono del usuario.

email: Dirección de correo electrónico del usuario.

**Bibliotecario**

Almacena información sobre los bibliotecarios que gestionan los préstamos de la biblioteca.

**Campos:**

id\_bibliotecario: Identificador único del bibliotecario (clave primaria).

nombre: Nombre del bibliotecario.

apellido: Apellido del bibliotecario.

dirección: Dirección del bibliotecario.

teléfono: Número de teléfono del bibliotecario.

**Préstamo**

Almacena información sobre los préstamos de libros realizados a los usuarios.

**Campos:**

id\_prestamo: Identificador único del préstamo (clave primaria).

id\_usuario: Identificador del usuario que realiza el préstamo (clave foránea).

id\_libro: Identificador del libro prestado (clave foránea).

id\_bibliotecario: Identificador del bibliotecario que registra el préstamo (clave foránea).

fecha\_prestamo: Fecha en que se realizó el préstamo.

fecha\_vencimiento: Fecha en que debe devolverse el libro.

fecha\_devolucion: Fecha en que se devolvió el libro (puede ser NULL si aún no se ha devuelto).

**AuditoriaPrestamo**

Almacena un registro de las acciones realizadas sobre los préstamos.

**Campos:**

id\_auditoria: Identificador único de la auditoría (clave primaria).

id\_prestamo: Identificador del préstamo asociado (clave foránea).

acción: Descripción de la acción realizada (por ejemplo, "préstamo registrado", "préstamo devuelto").

fecha: Fecha en que se realizó la acción.

**Diagrama Entidad Relacion**

A continuación se muestra el DER(Diagrama Entidad Relacion):



Relaciones entre Tablas:

**Tabla libro y tabla préstamo(1 a N)**: Un libro puede estar asociado a muchos prestamos, pero cada préstamo va a tener asociado un solo libro.

**Tabla usuario y tabla préstamo(1 a N):** Un usuario puede tener muchos prestamos, pero cada préstamo esta asociado a un usuario.

**Tabla bibliotecario y tabla préstamo(1 a N):** Un bibliotecario puede otorgar muchos prestamos, pero cada préstamo va a estar asociado a un solo bibliotecario.

**Tabla préstamo y tabla auditoriaPrestamo(1 a N):** Un préstamo puede tener muchas auditorias, pero cada auditoria va a estar asociada a un solo préstamo.

**Procedimiento y Transacción**

Este procedimiento sirve para registrar un nuevo préstamo, verificando que el libro esté disponible, en caso de que, si lo esté, iniciara una transacción la cual insertara un nuevo préstamo, actualizara el estado del libro a “prestado” y luego insertara un registro en la tabla “auditoriaPrestamo” con los datos del préstamo registrado.



Pasos:

Se crea el procedimiento llamado “registrar\_prestamo” y se le pasa como parámetro el id del libro, el id del usuario, y la fecha en la que se realiza el préstamo, a estos datos que ingresan se les asigna un alias para poder manipularlos más adelante.

Se declaran 3 variables, una llamada “libro\_prestado” que va a almacenar el estado del libro, obtenido desde la tabla “libro” y que puede ser “disponible” o “prestado”.

La segunda variable “v\_id\_bibliotecario” almacena el id del bibliotecario que en este caso se deja por defecto en 1.

La tercer variable v\_fecha\_vencimiento va a almacenar la fecha de vencimiento del préstamo que se calcula mas adelante.

Se hace una consulta SELECT a la tabla libro con el “id\_libro” que recibimos por parámetro, para almacenar el estado del libro en la variable que declaramos anteriormente.

Se evalúa con un IF el estado del libro almacenado en la variable, si el estado es “prestado” lanzara un mensaje con la alerta: “El libro ya está prestado”, sino continuara la ejecución en el ELSE.

Luego se inicia una transacción, en la cual de antemano desactivamos el “autocommit” que lo que hace es confirmar toda la transacción después de ejecutarse, esto se hace de manera manual al final de la transacción con el “COMMIT”.

Inicia la transacción con “STAR TRANSACTION”

En la variable “v\_fecha\_vencimiento” se inserta el vencimiento, con DATE\_ADD se calcula en base a la fecha del préstamo y se le suma un intervalo de 30 días.

Luego con todos los datos necesarios se hace la inserción del préstamo y se actualiza mediante un update el estado del libro a “prestado”.

Despues se hace una inserción a la tabla “auditoriaPrestamo” con el ultimo id insertado en la tabla “prestamo”, la acción de la auditoria y la fecha del préstamo.

Por ultimo si todas las operaciones fueron exitosas se confirman los cambios permanentemente con “COMMIT” y se vuelve a dejar por defecto el “autocommit”.

Se llama al procedimiento de la siguiente forma:



**Vista**

Esta vista sirve para ver los libros que están prestados actualmente y se quiere mostrar el id del préstamo, el título del libro y el nombre del usuario al cual se le presto. Los libros prestados son por lógica los que no tienen fecha de devolución.



Pasos:

Se crea la vista con “CREATE VIEW” y el nombre de la vista “vista\_prestamos\_actuales”.

Se hace la consulta SELECT desde la tabla préstamo, con el id del préstamo, el titulo del libro al cual se lo renombra para que se muestre en la consulta como “nombre\_libro”, el nombre y apellido del usuario que solicito el préstamo, con “CONCAT” se concatenan los dos campos para que se muestren en la consulta como una sola columna, y la fecha del préstamo. A todos los campos se les asigna un alias para poder manipularlos en los “JOIN” mas adelante.

Desde préstamo se hace un JOIN a la tabla libro mediante el id del libro para traer el nombre del libro.

Luego se hace un JOIN a la tabla usuario mediante el id del usuario para traer el nombre y apellido.

Por ultimo se pone la condición de que los prestamos que se quieren consultar solo los que la fecha de devolución de NULL o nulo, es decir, que no tengan fecha de devolución y por lógica, el libro sigue en préstamo.

**Trigger**

El trigger actualizar\_fecha\_devolucion se utiliza para automotizar la actualizacion de la fecha de devolucion en la tabla prestamo. Este trigger asegura que cada vez que se actualiza un prestamo, si se otorga una fecha de devolucion, se registre automaticamente la fecha actual como la fecha de devolucion

Pasos:

Se crea un trigger llamado "actualizar\_fecha\_devolucion" que se ejecuta antes de que se actualice la tabla "prestamos"



* Condicion del trigger:

IF NEW.fecha\_devolucion IS NOT NULL THEN: Verifica si la fecha de devolucion no es nula

* Accion del trigger:

SET NEW.fecha\_devolucion = CURDATE();: Si la condicion se cumple, establece la fecha de devolucion a la fecha actual