

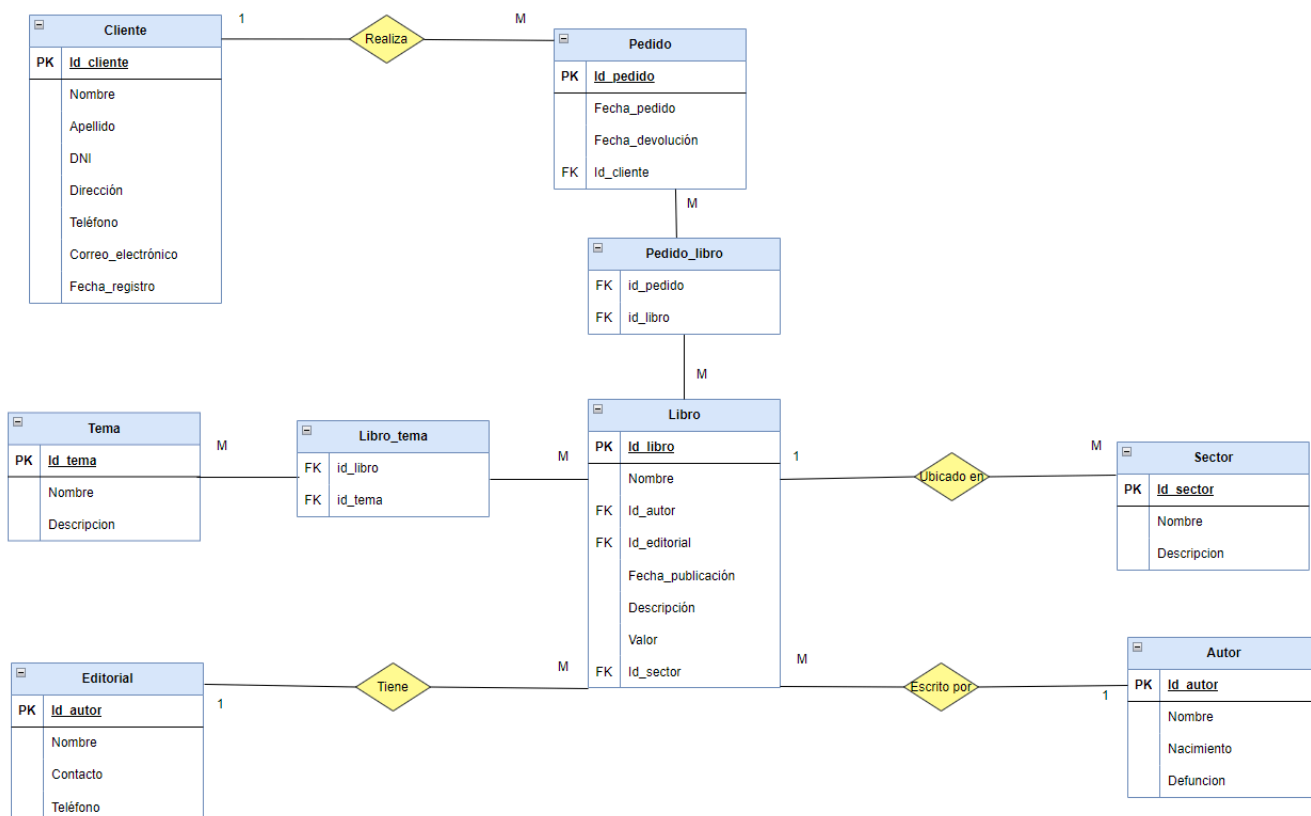
Biblioteca

En este documento se detalla una base de datos creada específicamente para una biblioteca. Se abordará tanto la problemática como la creación y descripción de la base de datos para esta. Posteriormente se creará un análisis de los datos obtenidos en la base.

El objetivo de la base de datos mostrada a continuación es servir de eje central para la organización de una biblioteca. La base proveerá a los empleados una vista detallada de los libros existentes en el establecimiento. De la misma manera, contará con una lista de clientes registrados, los cuales podrán realizar alquileres de los libros.

La problemática principal que trataremos de resolver con esta base de datos es la organización de una biblioteca donde la gente pueda alquilar libros. El establecimiento está compuesto por varios pisos y exhibidores donde se ponen los libros, incluyendo una bodega donde se almacenan. Los libros se ordenan en los exhibidores por géneros. Los clientes podrán alquilar los libros de su preferencia, con un plazo de devolución y un precio por el servicio. La biblioteca mantiene contacto con las editoriales, a las cuales les compra los libros.

La base de datos consta de 15 tablas las cuales 6 son para almacenar logs de un inserts, updates y deletes de las tablas cliente y pedido_libro. En la tabla cliente se almacenan las personas que quieren alquilar un libro, este alquiler es almacenado en la tabla pedido, asignándole una fecha de pedido y devolución. El cliente puede pedir más de un libro por lo que estos múltiples pedidos son almacenados en la tabla pedido_libro. Esta última almacena la ID del pedido y la de un libro. Los libros son almacenados en la tabla libro, que a su vez está conectada, a través de una FK a las tablas editorial, autor y sector. Como un libro puede abordar varios temas, la tabla tema almacena las posibles temáticas de libros en general y la tabla libro_tema almacena un libro con una de sus temáticas en específico.



Tablas

- Tabla Cliente: almacena los diversos clientes que se ingresen para alquilar libros.
- Tabla Pedido: almacena los pedidos realizados por clientes.
- Tabla Pedido_Libro: almacena id de libros alquilados en un pedido.
- Tabla Libros: almacena los libros existentes en la biblioteca.
- Tabla Libro_Tema: almacena id de temas pertenecientes a un libro.
- Tabla Tema: almacena temas pertenecientes a los libros existentes.
- Tabla Sector: almacena sectores en donde se almacenan los libros.
- Tabla Editorial: almacena editoriales pertenecientes a los libros.
- Tabla Autor: almacena autores pertenecientes a los libros.
- Tabla cliente_log_i: almacena los logs de inserts de las tabla cliente
- Tabla cliente_log_u: almacena los logs de updates de las tabla cliente
- Tabla cliente_log_d: almacena los logs de deletes de las tabla cliente
- Tabla pedido_libro_log_i: almacena los logs de inserts de las tabla cliente
- Tabla pedido_libro_log_u: almacena los logs de updates de las tabla cliente
- Tabla pedido_libro_log_d: almacena los logs de deletes de las tabla cliente

Excel con las tablas y variables:

<https://docs.google.com/spreadsheets/d/1H0xNnllna8tsxUOhO1HjuSpYmDc7imxVhwMaa7oLKA4/edit?usp=sharing>

Scripts de creación de tablas:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/scriptBiblioteca.sql

Datos

Scripts de inserción de datos:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/ScriptIns%2B%20Urquiza.sql

Vistas

ID_CLIENTE_ID_LIBRO: Vista que junta los ids de los clientes y los ids de los libros alquilados por estos, junto con las fechas de préstamo y devolución. Se hace uso de las tablas pedido_libro y pedido.

CLIENTE_LIBRO: Vista que junta los datos de clientes, sus libros alquilados, las fechas de préstamo y devolución. Se hace uso de los datos de las tablas cliente y libro junto con la vista id_cliente_id_libro.

CANTIDAD_ALQUILERES_POR_LIBRO: Vista que muestra la cantidad de préstamos o alquileres por libro. Se hace uso de los datos de las tablas pedido_libro y libro.

CANTIDAD_LIBROS_POR_TEMA: Vista que muestra la cantidad de libros por tema. Se hace uso de los datos de las tablas libro_tema y tema.

LIBRO_INFORMACION_COMPLETA: Vista que muestra el nombre del libro, del autor, de la editorial, fecha de publicación, valor monetario del libro y sector donde está ubicado. Se hace uso de las tablas libro, autor y sector.

CANTIDAD_ALQUILERES_POR_CLIENTE: Vista que muestra la cantidad de alquileres que realizó un cliente. Se hace uso de los datos de la tabla cliente y la vista id_cliente_id_libro.

LIBROS DE CLIENTE: Vista que junta los datos de clientes, sus libros alquilados, las fechas de préstamo y devolución. Se hace uso de los datos de las tablas pedido_libro, pedido, libro y cliente.

Scripts de creación de vistas:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/ScriptVis%2BUrquiza.sql

Funciones

Calcular costo alquiler(id int,dias int): esta función se encarga de calcular los costos de alquiler de un libro haciendo uso del dato "valor_libro" de la tabla libro. El parámetro "id" es el id del libro a alquilar y el dato "dias" es la cantidad de días que esto va ser alquilado.

Calcular cantidad libros por usuario(usuario varchar(50)): esta función se encarga de calcular la cantidad de libros alquilados por un cliente. Hace uso del dato "alquileres" de la vista cantidad_alquileres_por_cliente. El parámetro "nombre" hace referencia al nombre con el cual fue ingresado el cliente en la base de datos.

Scripts de creación de funciones:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/Formato%20%2BUrquiza.sql

Stored Procedures

Ordenar_clientes(IN campo char(20),in orden char(20)): se encarga de ordenar la tabla cliente. El primer parámetro indica el campo por el cual se ordenará y el segundo indica tipo de orden, ascendente(asc) o descendente(desc).

Agregar_cliente(in nombre VARCHAR(20),in apellido VARCHAR(20),in dni VARCHAR(10),in direccion VARCHAR(50),in telefono VARCHAR(15),in correo_electronico VARCHAR(50),in fecha_registro DATETIME,in nacimiento DATETIME): Se encarga de agregar registros a la tabla cliente, los parámetros siguen el siguiente orden: nombre, apellido, dni, dirección, teléfono, correo electrónico, fecha registro y nacimiento.

Scripts de creación de stored procedures:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/Stored%2BUrquiza.sql

Triggers

Cliente_before_update: inserta los clientes modificados en la tabla cliente en la tabla cliente_log_u junto con la hora, fecha y el usuario que realizó la acción.

Cliente_before_delete: inserta los clientes eliminados en la tabla cliente en la tabla cliente_log_d junto con la hora, fecha y el usuario que realizo la accion.

Cliente_after_insert : inserta los clientes ingresados en la tabla cliente en la tabla cliente_log_i junto con la hora, fecha y el usuario que realizó la acción.

Pedido_libro_before_insert : Inserta las filas insertadas en la tabla pedido_libro en la tabla pedido_libro_log_i junto con la hora, fecha y el usuario que realizó la accion.

Pedido_libro_before_update : Inserta las filas modificadas en la tabla pedido_libro en la tabla pedido_libro_log_u junto con la hora, fecha y el usuario que realizó la accion.

Pedido_libro_before_delete : inserta las filas eliminadas en la tabla pedido_libro en la tabla pedido_libro_log_d junto con la hora, fecha y el usuario que realizó la acción.

Scripts de creación de triggers:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/Triggers%2BURquiza.sql

Usuarios

Se crearon dos tipos de usuarios:

Usuario_Nivel_3: al cual se le dieron permisos de lectura sobre las tablas del Schema Biblioteca.

Usuario_Nivel_2: al cual se le dieron permisos de lectura, escritura y modificación sobre todas las tablas del Schema Biblioteca.

Scripts de creación de usuarios:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/Sentencias%2BURquiza.sql

Transacciones

Se utilizan transacciones para gestionar los cambios en la base de datos y tener mayor seguridad si comete un error. En este caso se usaron dos.

En la primera transacción se elimina un algunos registros y en la segunda transacción se agrega nuevos registros haciendo uso de savepoints.

Script de transacciones:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/TCL%2BURquiza.sql

BackUp y Restauración

Se creó un backup incluyendo solo los datos, dejando de lado la estructura.

Script:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/Backup%2BURquiza.sql

Informes generados en base a la información almacenada

En el siguiente análisis se grafica la cantidad de veces que fue alquilado cada libro.

Excel con el análisis a una tabla del proyecto:

https://docs.google.com/spreadsheets/d/1fpfC76yFH_J1cnRF2kDOliEd2LGVTm77jjh_wvIWAJk/edit?usp=sharing

Scripts de creación de la estructura y datos del Proyecto completo:

https://github.com/julio-urquiza/Proyecto_Final_CoderHouse/blob/main/ScriptGeneral.sql

Tecnologías y programas utilizados

Para la creación de la base de datos mostrada en este documento se usó MySQL Workbench 8.0, los diagramas entidad-relación se realizaron en draw.io (<https://www.drawio.com/>), se utilizó hoja de cálculo de Google para describir y diagramar las columnas de las tablas, también se utilizó Github (<https://github.com/>) para crear el repositorio en donde se encuentran almacenados los scripts creados a lo largo de este documento. Los conocimientos adquiridos para la realización de este proyecto fueron brindados gracias al curso de SQL de CoderHouse.