



SOLUCIÓN PROPUESTA PROYECTO N°2

Descripción del Problema para Modelar

Se desea crear un sistema de base de datos para un sistema sencillo de una biblioteca. Es importante llevar un registro de los autores, de los cuales se almacenará un código o identificador y su nombre. El sistema de bibliotecas guarda información de los usuarios, de los cuales interesa guardar un código, nombre y el teléfono, el cual debe permitir el uso de guiones y hasta un máximo de 12 caracteres en la cadena).

La información de los libros también debe ser almacenada en la base de datos, para lo cual se necesita conocer un identificador, el título del libro, su precio en decimal con el detalle de verificación (se tiene que validar que el precio esté en el rango de entre 1.00 y 150.00, inclusive), el formato del libro (debe usarse la nomenclatura “pd” para portada dura, “pb” para portada blanda y “cd” para libro digital y se debe verificar que solamente se acepten esta nomenclatura). Se sabe que un usuario puede solicitar los libros, por lo que la información de código del usuario debe estar asociada al Libro.

Para este caso, un libro solo puede ser pedido por un usuario, aunque se tienen varias copias de los libros más solicitados, aunque los usuarios pueden solicitar más de un libro. Adicionalmente, se necesita mantener un registro de qué autor escribió un determinado libro, sabiendo que un autor puede escribir varios libros, y que los libros pudieran tener más de un autor.

Restricciones de integridad

A continuación, se describen las tablas que se solicitan, con sus respectivos atributos (columnas), llaves, restricciones y referencias a otras tablas:

Autor

- A_ID (entero, llave primaria)
- Nombre (caracter (20), no puede ser nulo)

Libro

- L_ID (entero, llave primaria)
- Título (caracter (25), no puede ser nulo)
- Precio (decimal (5,2), no puede ser nulo. Se tiene que verificar que el previo sea mayor o igual que 1.00 pero menor o igual que 150.00)
- Formato (caracter (2), solo puede ingresarse tres valores: td, tb, cd)
- U_ID (llave foránea)

Autor_de_Libro

- ID_A (entero, llave foránea)
- ID_L (entero, llave foránea)
- La llave primaria es compuesta (ID_A, ID_L)

Usuario

- ID_U (entero, llave primaria)
- Nombre (caracter (20)), no puede ser nulo
- Teléfono (caracter (12))

Modelo Entidad Relación Propuesto

Tomando en cuenta la narración del problema a modelar se crea el modelo entidad relación con los siguientes elementos:

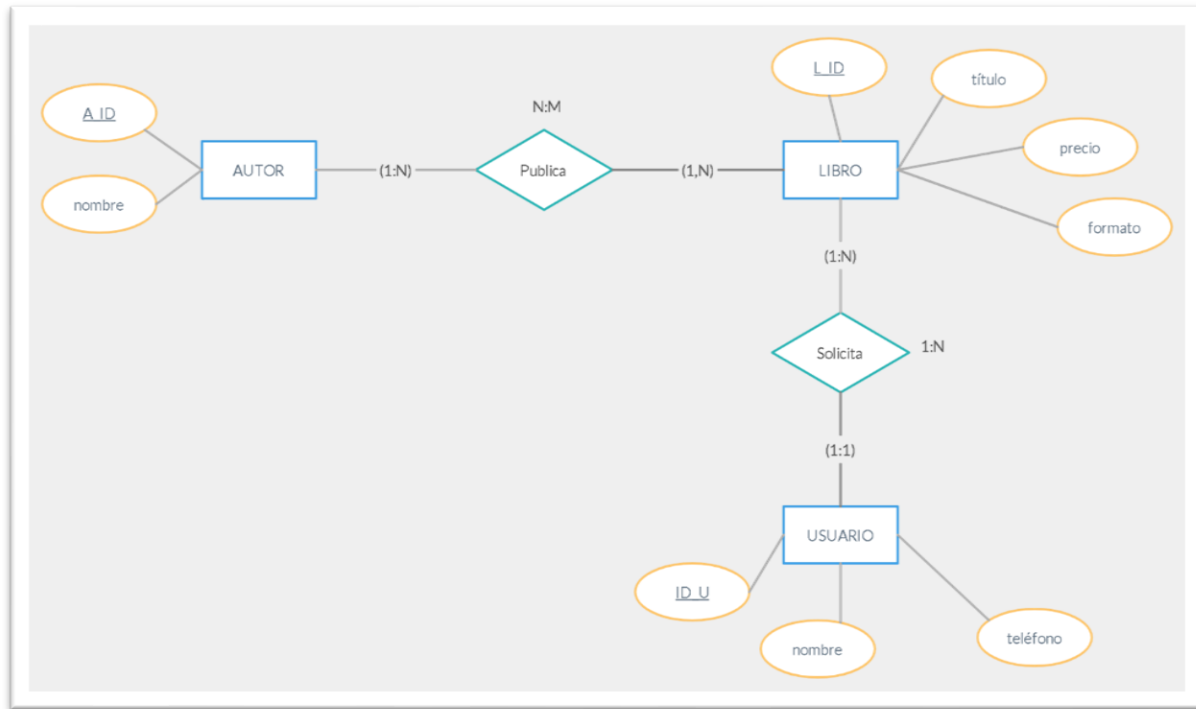
Entidades:

- Autor
- Libro
- Usuario

Relaciones:

- Escribe
- Solicita

A continuación, se presenta el diagrama Entidad Relación respectivo. Para ello se está usando la herramienta en línea <https://creatly.com/es/home/>



MODELO RELACIONAL:

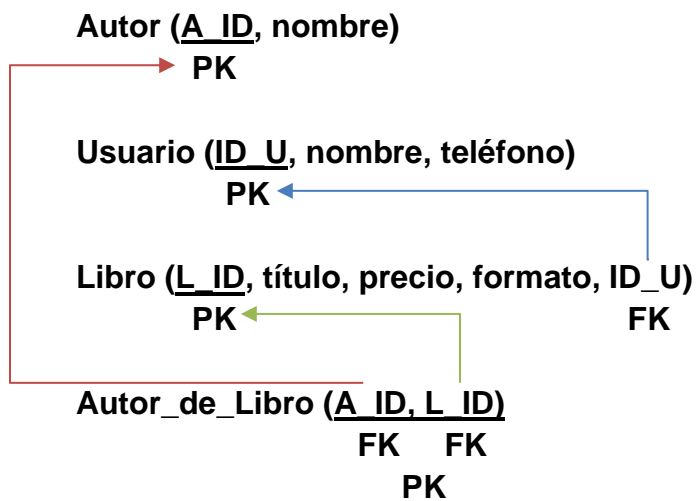


Diagrama de Base de Datos

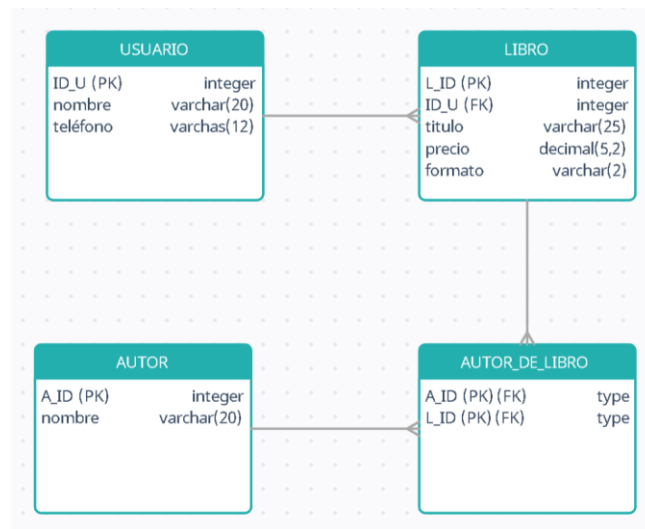


Diagrama generado en Creately

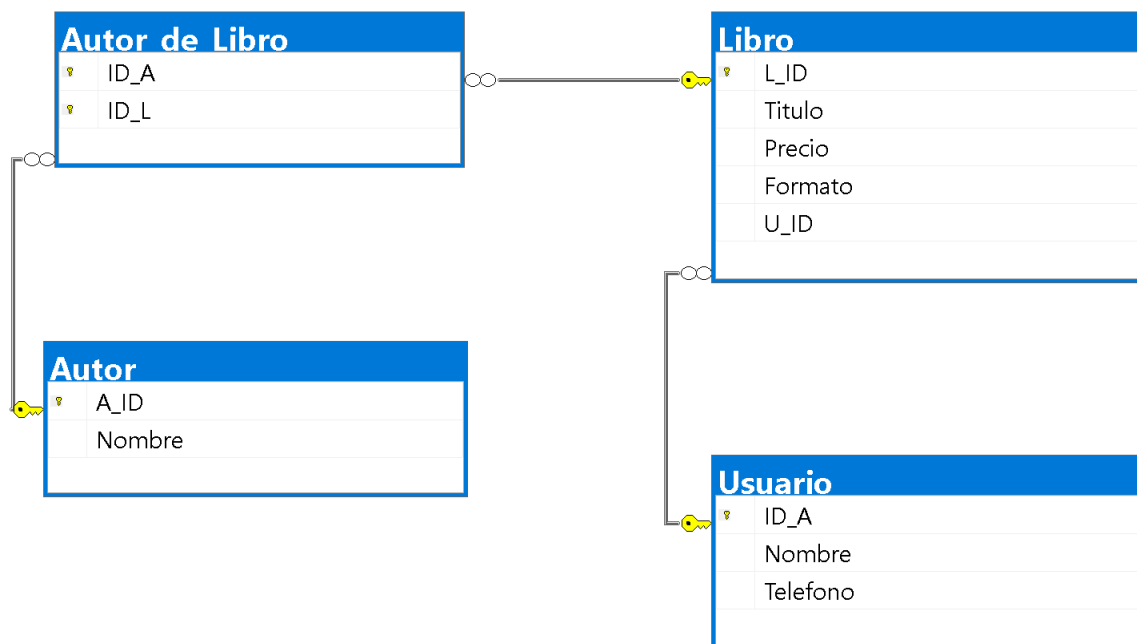


Diagrama Generado en MS SQL Server

SCRIPT DE CREACIÓN EN MS SQL SERVER

```
--SCRIPT DE CREACIÓN DEL PROYECTO BIBLIOTECA

--Se crea una nueva base de datos para el proyecto
create database Biblioteca_Proj

-- Proceso de creación de las tablas y carga de los datos
--Se debe crear en primer lugar las tablas que son sencillas y no tienen una llave
foranea en su estructura

--Creación de la tabla Autor
Create table Autor
(
A_ID Integer,
Primary Key(A_ID), --A_ID es llave primaria
Nombre varchar(20) not null
);

-- Introduciendo valores en tabla Autor. La inserción de los valores debe hacerse en el
orden en que aparecen en la estructura de la tabla
insert into Autor values (101, 'Karla Arosemena');
insert into Autor values (102, 'Doris Gutiérrez');
insert into Autor values (103, 'Vanessa Castillo');
insert into Autor values (104, 'Víctor Fuentes');
insert into Autor values (105, 'Nicolás Samaniego');
insert into Autor values (106, 'Dilsa Vergara');

select * from Autor --Visualiza la tabla respectiva para comprobar las inserciones

-- Creación de la tabla Usuario
Create table Usuario
(
ID_A Integer,
Primary Key (ID_A), /*ID_A es la llave primaria*/
Nombre varchar(20) not null,
Telefono varchar(12),
);

/*Insertando valores en la tabla Usuario*/
insert into Usuario values (501, 'Antonio Tello', '67564534');
insert into Usuario values (502, 'Diana Arosemena', '62554463');
insert into Usuario values (503, 'Milagros Díaz', '2533253');
insert into Usuario values (504, 'Nereida Muñoz', '2210210');
insert into Usuario values (505, 'Javier Peralta', '64001234');

select * from Usuario --Visualiza la tabla respectiva para comprobar las inserciones
```

```

/*Se crea ahora la tabla Libro. Esta tabla usa una llave foránea que hace referencia a
Usuario.
Note las opciones de verificación de valores con la cláusula CHECK*/
Create table Libro
(
L_ID Integer primary Key,
Titulo varchar (25) not null,
Precio decimal(5,2) not null check (Precio >= 1.00 and Precio <= 150.00),
Formato varchar(2) check (Formato in ('td', 'tb', 'cd')),
U_ID Integer not null Foreign Key (U_ID) references Usuario(ID_A), /*En este caso se le
ha dado un nombre diferente al de la tabla original*/
);

/*Se procede a insertar los valores en la tabla Libro. Note que el valor a insertar en
U_ID que es de la tabla Usuario (ID_A)
debe coincidir con algunos de los códigos que ya se introdujeron. El sistema no permite
que introduzca un valor que no exista.
De esta forma se garantiza la integridad de los datos */
insert into Libro values (1001, 'Base de Datos', 78.95, 'td', 503);
insert into Libro values (1002, 'Redes Inalámbricas', 65.49, 'cd', 505);
insert into Libro values (1003, 'Inteligencia de Negocios', 118.95, 'tb', 501);
insert into Libro values (1004, 'Machine Learning', 139.99, 'td', 504);
insert into Libro values (1005, 'Data Visualization', 145.63, 'cd', 502);
insert into Libro values (1006, 'Web Development', 115.75, 'tb', 502); /*Note que el usuario
502 tiene 2 libros registrados*/

select * from Libro --Visualiza la tabla respectiva para comprobar las inserciones

/*Ahora se procede a crear la tabla Autor_de_Libro que representa a quiénes han escrito
los libros.
Se introducen valores en donde un libro tendrá más de un autor, y también libros que son
escritos por el mismo autor.
Los valores que inserte en esta tabla deben coincidir con las llaves que ya usó debido a
que la llave es compuesta
y cada una de ellas por separado es una llave foránea */

Create table Autor_de_Libro
(
ID_A Int Not Null references Autor(A_ID), --references enlaza la FK con la tabla
respectiva
ID_L Int Not Null references Libro(L_ID),
primary key(ID_A, ID_L) -- La PK es compuesta, tiene dos atributos
);

--Insertando valores en la tabla Autor_de_Libro
insert into Autor_de_Libro values (106, 1001);
insert into Autor_de_Libro values (103, 1002);
insert into Autor_de_Libro values (102, 1003);
insert into Autor_de_Libro values (105, 1004);
insert into Autor_de_Libro values (104, 1005);
insert into Autor_de_Libro values (106, 1005); --Este libro es escrito también por el auor
106 (dos autores)
insert into Autor_de_Libro values (101, 1006);
insert into Autor_de_Libro values (102, 1006); --Este libro es escrito también por el auor
102 (dos autores)

select * from Autor_de_Libro --Visualiza la tabla respectiva para comprobar las
inserciones

```