

PROYECTO BIBLIOTECA

Autores: David Melgarejo, Sergio Ruiz, Juan de Dios Cáceres

Profesor: Francisco Martínez-Echeverría

Curso: 1º CFGS DAW

Asignatura: BD

Fecha: 13-Junio-2020

ÍNDICE

1. ENUNCIADO Y SUPOSICIONES	3
1.1. SUPOSICIONES	4
2. DIAGRAMA E/R	5
3. PASO A TABLAS	6
4. NORMALIZACIÓN	7
5. INGENIERÍA INVERSA	8
6. SCRIPT SQL	9
6.1. CREATE TABLE E INTEGRIDAD REFERENCIAL	9
6.2. DROP	13
7. OPERACIONES DML	14
7.1. SELECT	14
7.2. INSERT	16
7.3. UPDATE	26
7.4. DELETE	26
8. OPERACIONES CON PL/SQL:	26
8.1. CURSORES	26
8.2. TRIGGERS	29
8.3. PROCEDIMIENTOS Y FUNCIONES	34

1. ENUNCIADO Y SUPOSICIONES

Se propone realizar una aplicación enfocada en tres áreas: inventario de recursos gestionados por la biblioteca, registro de usuarios, y por último préstamos y devoluciones de recursos. Tras un par de jornadas de conversaciones con los técnicos, se recogen las siguientes especificaciones:

La biblioteca gestiona recursos bibliográficos, audiovisuales y electrónicos materializados sobre cualquier tipo de soporte. Debemos saber cuál es el material que da soporte a cada recurso, por ejemplo, libros, cintas de audio, películas, elementos multimedia, mapas, videos y otros.

Por otra parte, los recursos de la biblioteca se dividen en colecciones (Novela, Ficción, Ensayo, Juvenil, Periódicos, Vídeos). Que un recurso pertenezca a una colección determina el período máximo de préstamo y la penalización por cada día de retraso en su devolución.

Los recursos bibliográficos constituyen una de las entidades principales de la aplicación. Como hemos visto, cada recurso pertenece a una colección y es de un soporte de material específico. Cualquier recurso tiene los siguientes atributos: título, autores, declaración de responsabilidad de los contenidos.

Los ejemplares de los recursos bibliográficos constituyen los objetos de préstamo de la biblioteca. Uno de los libros de “War and Peace” de la biblioteca es un ejemplar del recurso “War and Peace” en soporte “Libro” de la colección “Novela”, de la misma manera que un compact disc de “Thriller” es una copia del recurso del mismo nombre en formato CD. Cada ejemplar se identifica por un código de barras único que lo diferencia de cualquier otro ejemplar de la biblioteca.

Un ejemplar puede estar en diferentes estados (Disponible, Prestado, Perdido, Dañado, Pedido, etc.). En función de su estado será posible su préstamo. Debe guardarse un historial de todos los estados por los que ha pasado un ejemplar a lo largo de su historia.

Los socios son los usuarios de la biblioteca. Cada miembro se identifica de manera única por un código de barras que está dibujado en su carnet de socio. De cada miembro se dispone de sus datos personales: nombre, dirección, teléfono, dirección electrónica.

La biblioteca clasifica sus socios en diferentes perfiles. Por lo general, estos perfiles son “Juvenil” y “Adulto”, aunque como es lógico, de una biblioteca a otra puede haber más perfiles de clasificación. Cada perfil tiene un límite de penalizaciones permitidas. Al alcanzar ese límite el socio pierde cualquier posibilidad de solicitar ningún recurso en préstamo.

En base al perfil de clasificación del socio de la biblioteca y al material que da soporte a un recurso se establece el tiempo límite de préstamo y el número de renovaciones permitidos. Por ejemplo, cualquier vídeo solicitado por un socio juvenil se podrá ser prestado por un periodo máximo de 10 días sin posibilidad de renovación.

Cuando no queda disponible ningún ejemplar de un recurso determinado y éste es solicitado en préstamo, es posible realizar una reserva sobre él. Cuando el ejemplar ha sido reservado, el usuario que lo tiene en préstamo pierde la posibilidad de renovarlo y se ve en la obligación de devolverlo en la fecha pactada.

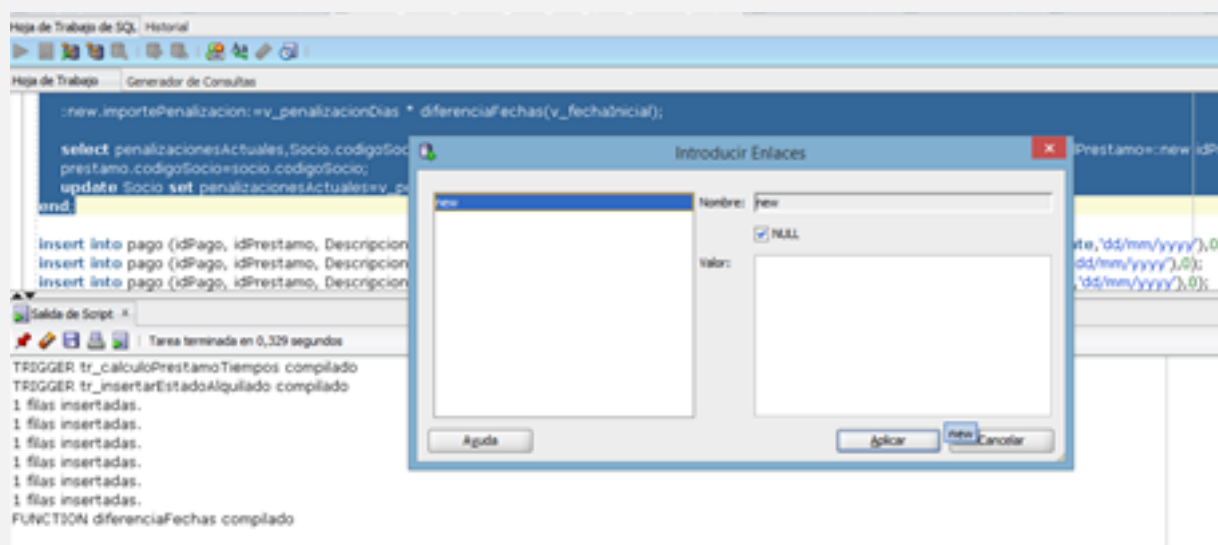
Se deben registrar los pagos de cuotas o penalizaciones que realicen los socios por medio de los distintos tipos de transacción aceptados por la biblioteca (efectivo, tarjeta de crédito, etc.) anotando el importe, la fecha, la descripción y el tipo de transacción.

Por último, se mantendrá un fichero con los empleados de la biblioteca que deben manejar esta aplicación informática. En este registro contendrá los códigos de usuario y contraseña para autenticarse en el aplicativo. Además, tendrá la información para autorizar en los distintos de la aplicación en los que el empleado podrá actuar. Por otra parte, cuando un empleado haga una anotación en las tablas de recursos, socios, usuarios o pagos debe quedar reflejado.

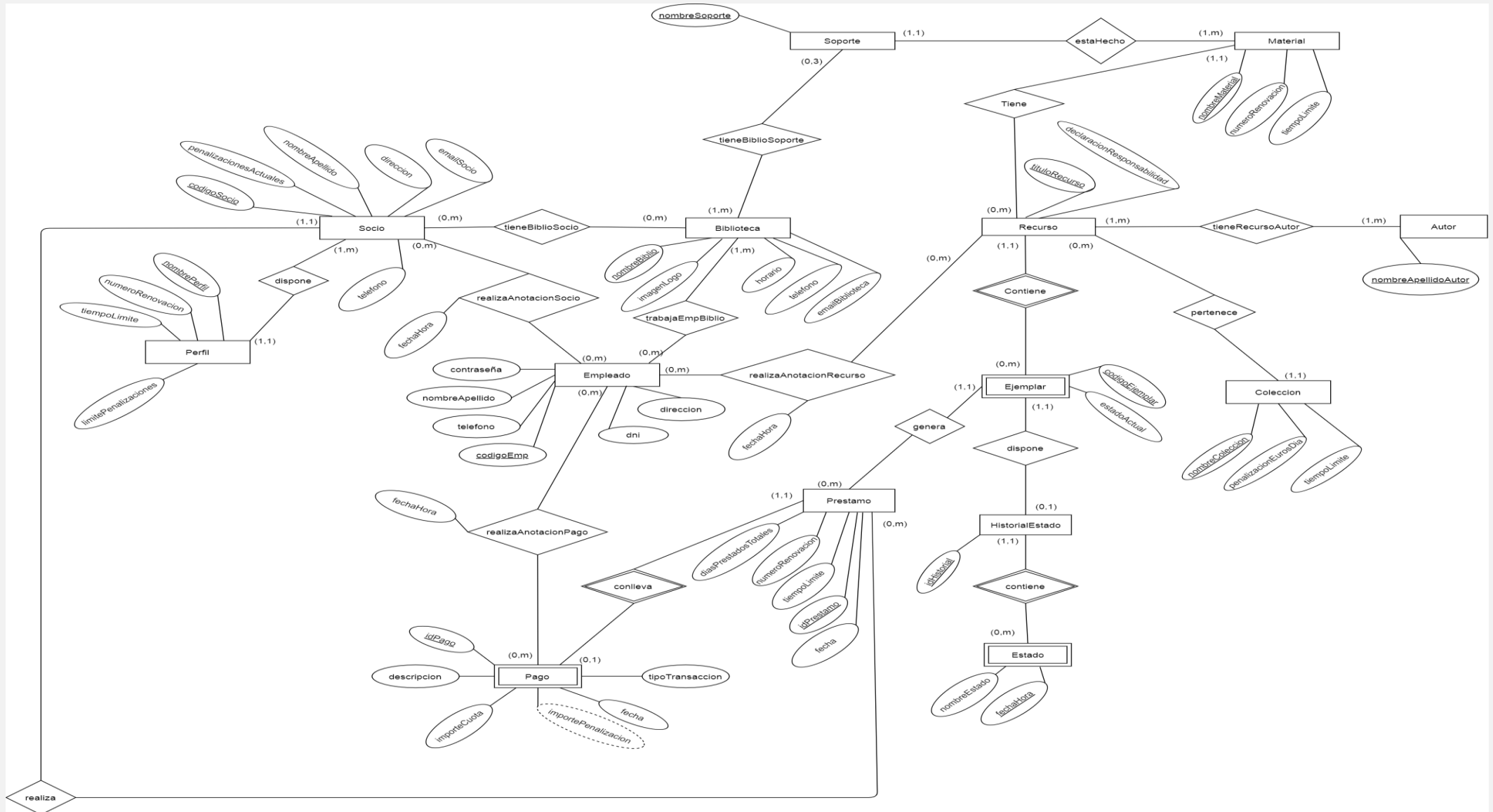
La aplicación está pensada para una red de bibliotecas. Por cada biblioteca se guarda: el nombre, una imagen con el logo, el horario, el teléfono y la dirección web de internet. Los ejemplares de los recursos bibliográficos están asignados a una biblioteca. Sin embargo los socios lo son de la red de bibliotecas y pueden usar los servicios de cualquiera de ellas.

1.1. Suposiciones

- La primary key de recurso es compuesta por tituloRecurso y nombreMaterial, ya que el tituloRecurso podría ser el mismo para diferentes materiales (por ejemplo libro - DVD, etc.).
- En préstamo el atributo tiempoLimite se calcula con la media de los atributos limiteMaterial (de la tabla material) y limitePerfil (de la tabla perfil), el atributo numeroRenovacion se obtiene de la media entre renovacionMaterial (tabla material) y renovacionPerfil (tabla perfil), y el atributo diasPrestadosTotales se calcula con el trigger tr_diasPrestadosTotales.
- El atributo importePenalizacion es derivado ya que se calcula cada vez con el trigger tr_diasPrestadosTotales.
- En pago, la cardinalidad con préstamo (0,1) se refiere a la existencia de la realización de un pago o no.
- La cardinalidad de Ejemplar hacia HistorialEstado es (0,1) porque en el momento de introducir un ejemplar no se dispone de historial (se crea después).
- Los códigos de empleado son a nivel de todas las bibliotecas.
- idPago (tabla pago) es por cada préstamo.
- Posible mensaje de aceptar esto.



2. DIAGRAMA E/R



3. PASO A TABLAS

Leyenda

Negrita y subrayado simple → PK

Subrayado doble → PK y FK

Subrayado discontinuo → Fk

Socio (**codigoSocio**, nombreApellido, direccion, telefono, emailSocio, penalizacionesActuales, nombrePerfil)

Perfil (**nombrePerfil**, numeroRenovacion, tiempoLimite, limitePenalizaciones)

tieneBiblioSocio (**nombreBiblio, codigoSocio**)

Biblioteca (**nombreBiblio**, imagenLogo, horario, telefono, emailBiblioteca)

Soporte (**nombreSoporte**)

tieneBiblioSoporte(**nombreBiblio, nombreSoporte**)

Material (**nombreMaterial**, numeroRenovacion, tiempoLimite, nombreSoporte)

Recurso (**tituloRecurso, nombreMaterial**, declaracionResponsabilidad, nombreColeccion)

tieneRecursoAutor (**tituloRecurso, nombreMaterial, nombreApellidoAutor**)

Autor (**nombreApellidoAutor**)

Coleccion (**nombreColeccion**, penalizacionEurosDia, tiempoLimite)

Ejemplar (**codigoEjemplar, tituloRecurso, nombreMaterial**)

HistorialEstado (idHistorial, codigoEjemplar, tituloRecurso, nombreMaterial)

Estado (fechaHora, idHistorial, nombreEstado)

Prestamo (idPrestamo, fecha, tiempoLimite, diasPrestadosTotales, numeroRenovacion, codigoEjemplar, tituloRecurso, nombreMaterial, codigoSocio)

Pago (idPago, idPrestamo, descripcion, tipoTransaccion, fecha, importeCuota, importePenalizacion)

trabajaEmpBiblio (nombreBiblio, codigoEmp)

realizaAnotacionSocio (codigoSocio, codigoEmp, fechaHora)

Empleado (codigoEmp, contraseña, nombreApellido, telefono, dirección, dni)

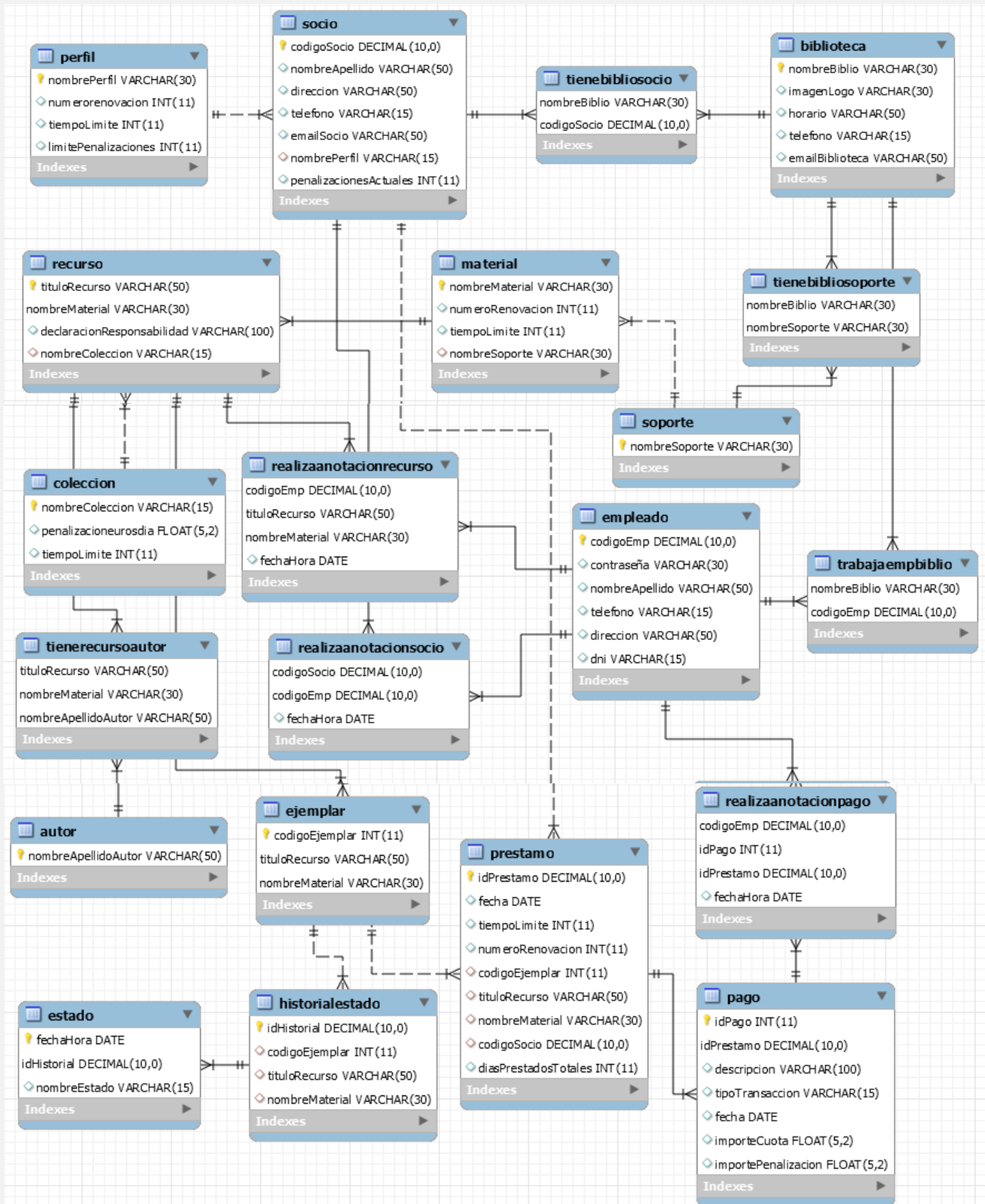
realizaAnotacionRecurso (codigoEmp, tituloRecurso, nombreMaterial, fechaHora)

realizaAnotacionPago (codigoEmp, idPago, idPrestamo, fechaHora)

4. NORMALIZACIÓN

Todas las tablas ya se encuentran normalizadas en 3FN.

5. INGENIERÍA INVERSA



6. SCRIPT SQL

6.1. Create table e integridad referencial

```
create table Perfil(  
    nombrePerfil varchar2(30) primary key,  
    numeroRenovacion int,  
    tiempoLimite int,  
    limitePenalizaciones int  
);  
  
create table Socio(  
    codigoSocio numeric primary key,  
    nombreApellido varchar2(50),  
    direccion varchar2(50),  
    telefono varchar2(15),  
    emailSocio varchar2(50),  
    nombrePerfil varchar2(15),  
    penalizacionesActuales int,  
    foreign key(nombrePerfil) references Perfil(nombrePerfil)  
);  
  
create table Biblioteca(  
    nombreBiblio varchar2(30) primary key,  
    imagenLogo varchar2(30),  
    horario varchar2(50),  
    telefono varchar2(15),  
    emailBiblioteca varchar2(50)  
);  
  
create table tieneBiblioSocio(  
    nombreBiblio varchar2(30),  
    codigoSocio numeric,  
    primary key(nombreBiblio,codigoSocio),  
    foreign key (nombreBiblio) references Biblioteca (nombreBiblio),  
    foreign key (codigoSocio) references Socio (codigoSocio)  
);
```

```
create table Soporte(  
    nombreSoporte varchar2(30) primary key  
);  
  
create table tieneBiblioSoporte(  
    nombreBiblio varchar2(30),  
    nombreSoporte varchar2(30),  
    primary key (nombreBiblio,nombreSoporte),  
    foreign key (nombreBiblio) references Biblioteca (nombreBiblio),  
    foreign key (nombreSoporte) references Soporte (nombreSoporte)  
);  
  
create table Material(  
    nombreMaterial varchar2(30) primary key,  
    numeroRenovacion int,  
    tiempoLimite int,  
    nombreSoporte varchar2(30),  
    foreign key (nombreSoporte) references Soporte (nombreSoporte)  
);  
  
create table Coleccion(  
    nombreColeccion varchar2(15) primary key,  
    penalizacioneuropea number(5,2),  
    tiempoLimite int  
);  
  
create table Recurso(  
    tituloRecurso varchar2(50),  
    nombreMaterial varchar2(30),  
    declaracionResponsabilidad varchar2(100),  
    nombreColeccion varchar2(15),  
    primary key(tituloRecurso,nombreMaterial),  
    foreign key (nombreMaterial) references Material (nombreMaterial),  
    foreign key (nombreColeccion) references Coleccion (nombreColeccion)  
);  
  
create table Autor(  
    nombreApellidoAutor varchar2(50) primary key  
);
```

```
create table tieneRecursoAutor(  
    tituloRecurso varchar2(50),  
    nombreMaterial varchar2(30),  
    nombreApellidoAutor varchar2(50),  
    primary key (tituloRecurso,nombreMaterial,nombreApellidoAutor),  
    foreign key (tituloRecurso,nombreMaterial) references Recurso (tituloRecurso,nombreMaterial),  
    foreign key (nombreApellidoAutor) references Autor (nombreApellidoAutor)  
);  
  
create table Ejemplar(  
    codigoEjemplar int,  
    tituloRecurso varchar2(50),  
    nombreMaterial varchar2(30),  
    primary key (codigoEjemplar,tituloRecurso,nombreMaterial),  
    foreign key (tituloRecurso,nombreMaterial) references Recurso (tituloRecurso,nombreMaterial)  
);  
  
create table HistorialEstado(  
    idHistorial numeric primary key,  
    codigoEjemplar int,  
    tituloRecurso varchar2(50),  
    nombreMaterial varchar2(30),  
    foreign key (codigoEjemplar,tituloRecurso,nombreMaterial) references Ejemplar  
(codigoEjemplar,tituloRecurso,nombreMaterial)  
);  
  
create table Estado(  
    fechaHora date,  
    idHistorial numeric,  
    nombreEstado varchar2(15),  
    primary key (fechaHora,idHistorial),  
    foreign key (idHistorial) references HistorialEstado(idHistorial)  
);  
  
create table Prestamo(  
    idPrestamo numeric primary key,  
    fecha date,  
    tiempoLimite int,
```

```
numeroRenovacion int,
codigoEjemplar int,
tituloRecurso varchar2(50),
nombreMaterial varchar2(30),
codigoSocio numeric,
diasPrestadosTotales int,
foreign key (codigoEjemplar,tituloRecurso,nombreMaterial) references Ejemplar
(codigoEjemplar,tituloRecurso,nombreMaterial),
foreign key (codigoSocio) references Socio (codigoSocio)
);
create table Pago(
    idPago int,
    idPrestamo numeric,
    descripcion varchar2(100),
    tipoTransaccion varchar2(15),
    fecha date,
    importeCuota number(5,2),
    importePenalizacion number(5,2),
    primary key(idPago,idPrestamo),
    foreign key (idPrestamo) references Prestamo (idPrestamo)
);
create table Empleado(
    codigoEmp numeric primary key,
    contraseña varchar2(30),
    nombreApellido varchar2(50),
    telefono varchar2(15),
    direccion varchar2(50),
    dni varchar2(15)
);
create table trabajaEmpBiblio(
    nombreBiblio varchar2(30),
    codigoEmp numeric,
    primary key (codigoEmp,nombreBiblio),
    foreign key (codigoEmp) references Empleado (codigoEmp),
```

```
foreign key (nombreBiblio) references Biblioteca (nombreBiblio)
);
create table realizaAnotacionSocio(
    codigoSocio numeric,
    codigoEmp numeric,
    fechaHora date,
    primary key (codigoSocio, codigoEmp),
    foreign key (codigoSocio) references socio (codigoSocio),
    foreign key (codigoEmp) references Empleado (codigoEmp)
);
create table realizaAnotacionRecurso (
    codigoEmp numeric,
    tituloRecurso varchar2(50),
    nombreMaterial varchar2(30),
    fechaHora date,
    primary key (codigoEmp, tituloRecurso,nombreMaterial),
    foreign key (tituloRecurso,nombreMaterial) references Recurso (tituloRecurso,nombreMaterial),
    foreign key (codigoEmp) references Empleado (codigoEmp)
);
create table RealizaAnotacionPago (
    codigoEmp numeric,
    idPago int,
    idPrestamo numeric,
    fechaHora date,
    primary key (codigoEmp, idPago, idPrestamo),
    foreign key (codigoEmp) references Empleado (codigoEmp),
    foreign key (idPago,idPrestamo) references Pago (idPago,idPrestamo)
);
```

6.2. Drop

```
drop table perfil cascade constraints;
drop table socio cascade constraints;
drop table biblioteca cascade constraints;
drop table tienebibliosocio cascade constraints;
```

```
drop table soporte cascade constraints;
drop table tienebibliosoporte cascade constraints;
drop table material cascade constraints;
drop table coleccion cascade constraints;
drop table recurso cascade constraints;
drop table autor cascade constraints;
drop table tienerecursoautor cascade constraints;
drop table historialestado cascade constraints;
drop table ejemplar cascade constraints;
drop table estado cascade constraints;
drop table prestamo cascade constraints;
drop table pago cascade constraints;
drop table empleado cascade constraints;
drop table trabajaempbiblio cascade constraints;
drop table realizaanotacionsocio cascade constraints;
drop table realizaanotacionrecurso cascade constraints;
drop table realizaAnotacionPago cascade constraints;
drop sequence secuencia_socio;
drop sequence secuencia_HistorialEstado;
drop sequence secuencia_Prestamo;
drop sequence secuencia_Empleado;
```

7. OPERACIONES DML

7.1. Select

=> Obtener el id historial dado el código ejemplar, nombre y material.

```
select idHistorial from HistorialEstado where codigoEjemplar = 3 and tituloRecurso = 'Star Wars
Episode 3: Revenge of the Sith' and nombreMaterial = 'Blu-ray';
```

=> Calcula la diferencia entre una fecha dada y la fecha actual.

```
select to_date(to_char(sysdate,'dd/mm/yyyy')) - to_date(to_char(03/06/2020,'dd/mm/yyyy')) into
difference from dual;
```

=> Obtener la penalización euros/día y el tiempo límite de un préstamo dado su id.

```
select penalizacionEurosDia, prestamo.tiempoLimite from coleccion,prestamo,recurso,ejemplar where  
prestamo.idPrestamo = '1' and prestamo.tituloRecurso=ejemplar.tituloRecurso and  
prestamo.nombreMaterial = Ejemplar.nombreMaterial and  
prestamo.codigoEjemplar=Ejemplar.codigoEjemplar and recurso.tituloRecurso = ejemplar.tituloRecurso  
and recurso.nombreMaterial = Ejemplar.nombreMaterial and recurso.nombreColeccion =  
coleccion.nombreColeccion;
```

=> Obtener el código socio y sus penalizaciones dado un id préstamo.

```
select penalizacionesActuales, Socio.codigoSocio from Socio, Prestamo where prestamo.idPrestamo =  
'1' and prestamo.codigoSocio = socio.codigoSocio;
```

=> Obtener los estados de un ejemplar dado su código, título y material.

```
select nombreEstado from estado, historialEstado, ejemplar, prestamo where estado.idHistorial  
= historialEstado.idHistorial and historialEstado.codigoEjemplar = ejemplar.codigoEjemplar and  
historialEstado.tituloRecurso = ejemplar.tituloRecurso and historialEstado.nombreMaterial =  
ejemplar.nombreMaterial and ejemplar.codigoEjemplar = prestamo.codigoEjemplar and  
prestamo.tituloRecurso = ejemplar.tituloRecurso and prestamo.nombreMaterial =  
ejemplar.nombreMaterial and ejemplar.codigoEjemplar = 1 and ejemplar.tituloRecurso = 'Juego  
de tronos' and ejemplar.nombreMaterial = 'DVD';
```

=> Obtener las penalizaciones totales de todos los socios.

```
select prestamo.codigoSocio, socio.nombreApellido, sum(importePenalizacion) from pago,  
prestamo, socio where pago.idPrestamo = prestamo.idPrestamo and prestamo.codigoSocio =  
socio.codigoSocio group by prestamo.codigoSocio,socio.nombreApellido;
```

=> Obtener todos los títulos de blu-ray que han sido alquilados.

```
select distinct ejemplar.tituloRecurso from ejemplar, historialEstado, estado where  
ejemplar.codigoEjemplar = historialEstado.codigoEjemplar and historialEstado.idHistorial =  
estado.idHistorial and ejemplar.nombreMaterial = 'Blu-ray' and estado.nombreEstado =  
'alquilado';
```

=> Obtener el código, título y estado de los ejemplares pedidos en los últimos 6 meses.

```
select ejemplar.codigoEjemplar, ejemplar.tituloRecurso, estado.nombreEstado from ejemplar,
historialEstado, estado where ejemplar.codigoEjemplar = historialEstado.codigoEjemplar and
ejemplar.tituloRecurso = historialEstado.tituloRecurso and ejemplar.nombreMaterial =
historialEstado.nombreMaterial and historialEstado.idHistorial = estado.idHistorial and
estado.nombreEstado = 'pedido' and ejemplar.nombreMaterial = 'Libro' and estado.fechaHora
> (select add_months(trunc(sysdate,'MM'),-6) from dual);
```

=> Obtener el número de ejemplares que hay de un recurso dado.

```
select count(codigoEjemplar) into totales from Ejemplar where tituloRecurso = 'Harry Potter y
el misterio del principe' and nombreMaterial = 'Libro' group by tituloRecurso, nombreMaterial;
```

=> Obtener el inventario total de una biblioteca.

```
select count(ejemplar.tituloRecurso) from ejemplar, recurso, material, soporte,
tieneBiblioSoporte, biblioteca where biblioteca.nombreBiblio = 'Uhu' and
ejemplar.tituloRecurso = recurso.tituloRecurso and ejemplar.nombreMaterial =
recurso.nombreMaterial and recurso.nombreMaterial = material.nombreMaterial and
material.nombreSoporte = soporte.nombreSoporte and soporte.nombreSoporte =
tieneBiblioSoporte.nombreSoporte and tieneBiblioSoporte.nombreBiblio =
biblioteca.nombreBiblio;
```

7.2. Insert

```
insert into Perfil values ('Infantil',3,3,5);
```

```
insert into Perfil values ('Juvenil',3,4,4);
```

```
insert into Perfil values ('Adulto',3,5,4);
```

```
insert into Perfil values ('Estudiante',5,4,3);
```

```
insert into Perfil values ('Jubilado',4,4,5);
```

```
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Kaka Roto','Campo','1234','Kaka@email.com','Adulto',0);
```



```
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Vege Ta','Ciudad Oeste','2345','Vege@email.com','Adulto',0);
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Majin Bu','Ciudad
Oeste','3456','Majin@email.com','Jubilado',0);
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Hora Cio','Ciudad del
Norte','4567','Hora@email.com','Infantil',0);
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Oli Ver','Campo','5678','Oli@email.com','Juvenil',0);
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Ben Ji','Futbol','6789','Ben@email.com','Juvenil',0);
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Gusta Bo','Gta V','7890','Gusta@email.com','Infantil',0);
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Katia Shan','Pueblo
Paleta','8901','Katia@email.com','Estudiante',0);
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Kri Lin','Ciudad Oeste','9012','Kri@email.com','Estudiante',0);
insert into Socio(nombreApellido, direccion, telefono, emailSocio, nombrePerfil,
penalizacionesActuales) values ('Muten Roshi','Kame
House','0123','Muten@email.com','Jubilado',0);
```

```
insert into Biblioteca values('Ugr','ugr.jpg','08:00-20:00','123458679','ugr@correo.es');
insert into Biblioteca values('Uma','uma.png','08:30-20:30','296571234','uma@correo.es');
insert into Biblioteca values('Uhu','uhu.jpg','08:00-21:30','900555123','uhu@correo.es');
insert into Biblioteca values('Complutense','comp.bmp','17:00-
22:00','555662233','comp@correo.es');
insert into Biblioteca values('Publica de Granada','gr.jpg','15:00-
21:00','987654321','gr@correo.es');
```

```
insert into Soporte values('Bibliografico');
```

```
insert into Soporte values('Audiovisual');
```

```
insert into Soporte values('Electronico');
```

```
insert into tienebibliosoporte values('Uhu','Bibliografico');
```

```
insert into tienebibliosoporte values('Uhu','Audiovisual');
```

```
insert into tienebibliosoporte values('Uhu','Electronico');
```

```
insert into tienebibliosoporte values('Ugr','Bibliografico');
```

```
insert into tienebibliosoporte values('Ugr','Electronico');
```

```
insert into tienebibliosoporte values('Uma','Audiovisual');
```

```
insert into tienebibliosoporte values('Complutense','Electronico');
```

```
insert into tienebibliosoporte values('Publica de Granada','Bibliografico');
```

```
insert into TieneBibliosocio values('Uhu','5');
```

```
insert into TieneBibliosocio values('Complutense','3');
```

```
insert into TieneBibliosocio values('Uma','1');
```

```
insert into TieneBibliosocio values('Publica de Granada','4');
```

```
insert into TieneBibliosocio values('Ugr','2');
```

```
insert into TieneBibliosocio values('Uma','7');
```

```
insert into TieneBibliosocio values('Ugr','6');
```

```
insert into TieneBibliosocio values('Uhu','8');
```

```
insert into TieneBibliosocio values('Publica de Granada','9');
```

```
insert into Material values('Libro',4,4,'Bibliografico');
```

```
insert into Material values('Libro electronico',4,3,'Electronico');
```

```
insert into Material values('Casete',3,2,'Audiovisual');
```

```
insert into Material values('Blu-ray',3,2,'Audiovisual');
```

```
insert into Material values('DVD',2,3,'Audiovisual');
```

```
insert into Material values('CD',4,2,'Audiovisual');
```

```
insert into Material values('Mapa',2,3,'Bibliografico');
```

```
insert into Material values('Mapa electronico',2,2,'Electronico');
```

```
insert into Coleccion values('Novela',1.80,5);
```

```
insert into Coleccion values('Ficcion',1.50,5);
```

```
insert into Coleccion values('Drama',1.50,4);
```

```
insert into Coleccion values('Tragedia',1.50,4);
```

```
insert into Coleccion values('Ensayo',2.00,6);
```

```
insert into Coleccion values('Juvenil',1.30,4);
```

```
insert into Coleccion values('Periodicos',0.90,2);
```

```
insert into Coleccion values('Videos',2.25,3);
```

```
insert into Coleccion values('Musica',2.25,3);
```

```
insert into Recurso values('Star Wars Episode 3: Revenge of the Sith','Blu-ray','Derechos reservados al autor','Ficcion');
```

```
insert into Recurso values('La leyenda del samurai: 47 ronin','Blu-ray','Derechos reservados al autor','Tragedia');
```

```
insert into Recurso values('La leyenda del samurai: 47 ronin','DVD','Derechos reservados al autor','Tragedia');
```

```
insert into Recurso values('Juego de tronos','DVD','Derechos reservados al autor','Ficcion');
```

```
insert into Recurso values('Juego de tronos','Libro','Derechos reservados al autor','Ficcion');
```

```
insert into Recurso values('Harry Potter y el misterio del principe','Libro','Derechos reservados al autor','Ficcion');
```

```
insert into Recurso values('Harry Potter y el misterio del principe','Blu-ray','Derechos reservados al autor','Ficcion');
```

```
insert into Recurso values('Breaking Bad','DVD','Derechos reservados al autor','Ficcion');
```

```
insert into Recurso values('La Senda Jedi','Libro','Derechos reservados al autor','Ficcion');
```

```
insert into Recurso values('Prison Break','Blu-ray','Derechos reservados al autor','Ficcion');
```

```
insert into Recurso values('Marca','Libro','Derechos reservados al autor','Periodicos');
```

```
insert into Recurso values('El bosque de los arboles muertos','Libro','Derechos reservados al autor','Juvenil');
```

```
insert into Recurso values('The Best of 2008-2012','CD','Derechos reservados al autor','Musica');
```

```
insert into Autor values('Miyake Miyamoto');
```

```
insert into Autor values('Dor Mir');
```

```
insert into Autor values('Carlos Ruiz');
```

```
insert into Autor values('Sergio Roncero');
```

```
insert into Autor values('Paulinho Scofield');
```

```
insert into Autor values('Mourinho ¿Porque?');
```

```
insert into Autor values('Bry Perez');
```

```
insert into Autor values('Ana Alcolea');
```

```
insert into Autor values('Laura Gallego');
```

```
insert into Autor values('G.E.M.');
```

```
insert into TieneRecursoAutor values('La leyenda del samurai: 47 ronin','Blu-ray','Miyake Miyamoto');
```

```
insert into TieneRecursoAutor values('Star Wars Episode 3: Revenge of the Sith','Blu-ray','Dor Mir');
```

```
insert into TieneRecursoAutor values('La leyenda del samurai: 47 ronin','DVD','Miyake Miyamoto');
```

```
insert into TieneRecursoAutor values('Juego de tronos','DVD','Carlos Ruiz');
```

```
insert into TieneRecursoAutor values('Juego de tronos','Libro','Sergio Roncero');
```

```
insert into TieneRecursoAutor values('Harry Potter y el misterio del principe','Libro','Laura Gallego');
```

```
insert into TieneRecursoAutor values('Harry Potter y el misterio del principe','Blu-ray','Ana Alcolea');
```

```
insert into TieneRecursoAutor values('Breaking Bad','DVD','Bry Perez');
```

```
insert into TieneRecursoAutor values('La Senda Jedi','Libro','Mourinho ¿Porque?');
```

```
insert into TieneRecursoAutor values('Prison Break','Blu-ray','Paulinho Scofield');
```

```
insert into TieneRecursoAutor values('Marca','Libro','Mourinho ¿Porque?');
```

```
insert into TieneRecursoAutor values('El bosque de los arboles muertos','Libro','Ana Alcolea');
```

```
insert into TieneRecursoAutor values('The Best of 2008-2012','CD','G.E.M.');
```

```
insert into Ejemplar values(1,'Prison Break','Blu-ray');
insert into Ejemplar values(2,'Prison Break','Blu-ray');
insert into Ejemplar values(3,'Prison Break','Blu-ray');
insert into Ejemplar values(1,'El bosque de los arboles muertos','Libro');
insert into Ejemplar values(1,'Harry Potter y el misterio del principe','Libro');
insert into Ejemplar values(1,'Harry Potter y el misterio del principe','Blu-ray');
insert into Ejemplar values(2,'Harry Potter y el misterio del principe','Blu-ray');
insert into Ejemplar values(3,'Harry Potter y el misterio del principe','Blu-ray');
insert into Ejemplar values(1,'Marca','Libro');
insert into Ejemplar values(1,'Star Wars Episode 3: Revenge of the Sith','Blu-ray');
insert into Ejemplar values(2,'Star Wars Episode 3: Revenge of the Sith','Blu-ray');
insert into Ejemplar values(3,'Star Wars Episode 3: Revenge of the Sith','Blu-ray');
insert into Ejemplar values(1,'La leyenda del samurai: 47 ronin','DVD');
insert into Ejemplar values(1,'Juego de tronos','DVD');
insert into Ejemplar values(1,'La Senda Jedi','Libro');
insert into Ejemplar values(1,'The Best of 2008-2012','CD');
insert into Ejemplar values(1,'Breaking Bad','DVD');
```

```
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(3,'Star Wars
Episode 3: Revenge of the Sith','Blu-ray');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'Breaking
Bad','DVD');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'El bosque
de los arboles muertos','Libro');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'Star Wars
Episode 3: Revenge of the Sith','Blu-ray');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'Harry
Potter y el misterio del principe','Libro');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial)
values(1,'Marca','Libro');
```

```
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'Juego de
tronos','DVD');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(2,'Star Wars
Episode 3: Revenge of the Sith','Blu-ray');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'La
leyenda del samurai: 47 ronin','DVD');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'La Senda
Jedi','Libro');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'Prison
Break','Blu-ray');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'The Best
of 2008-2012','CD');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(2,'Prison
Break','Blu-ray');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(3,'Prison
Break','Blu-ray');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(1,'Harry
Potter y el misterio del principe','Blu-ray');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(2,'Harry
Potter y el misterio del principe','Blu-ray');
insert into HistorialEstado (codigoEjemplar, tituloRecurso, nombreMaterial) values(3,'Harry
Potter y el misterio del principe','Blu-ray');

insert into Estado values('15/02/2012',1,'disponible');
insert into Estado values('08/04/1999',2,'disponible');
insert into Estado values('11/04/1999',2,'alquilado');
insert into Estado values('13/04/1999',2,'disponible');
insert into Estado values('17/12/2006',3,'disponible');
insert into Estado values('20/12/2006',3,'alquilado');
insert into Estado values('27/12/2006',3,'deteriorado');
insert into Estado values('31/03/2007',4,'disponible');
```

```
insert into Estado values('05/04/2007',4,'alquilado');
insert into Estado values('06/04/2007',4,'deteriorado');
insert into Estado values('28/02/2007',5,'disponible');
insert into Estado values('01/11/2018',6,'disponible');
insert into Estado values('01/11/2019',6,'alquilado');
insert into Estado values('04/11/2019',6,'disponible');
insert into Estado values('12/06/2001',7,'disponible');
insert into Estado values('04/07/1998',8,'disponible');
insert into Estado values('08/11/2003',9,'disponible');
insert into Estado values('02/02/2020',10,'disponible');
insert into Estado values('03/06/2020',11,'pedido');
insert into Estado values('04/06/2020',11,'disponible');
insert into Estado values('08/04/2018',12,'disponible');
insert into Estado values('08/04/2018',13,'disponible');
insert into Estado values('03/05/2014',14,'disponible');
insert into Estado values('23/01/2015',15,'disponible');
insert into Estado values('23/01/2015',16,'disponible');
insert into Estado values('23/01/2020',17,'pedido');
```

```
insert into Prestamo (fecha, codigoEjemplar, tituloRecurso, nombreMaterial, codigoSocio)
values ('03/06/2020',1,'Juego de tronos','DVD',3);
insert into Prestamo (fecha, codigoEjemplar, tituloRecurso, nombreMaterial, codigoSocio)
values ('05/06/2020',1,'Breaking Bad','DVD',2);
insert into Prestamo (fecha, codigoEjemplar, tituloRecurso, nombreMaterial, codigoSocio)
values ('03/06/2020',1,'The Best of 2008-2012','CD',5);
insert into Prestamo (fecha, codigoEjemplar, tituloRecurso, nombreMaterial, codigoSocio)
values ('04/06/2020',1,'El bosque de los arboles muertos','Libro',9);
insert into Prestamo (fecha, codigoEjemplar, tituloRecurso, nombreMaterial, codigoSocio)
values ('07/06/2020',3,'Star Wars Episode 3: Revenge of the Sith','Blu-ray',1);
insert into Prestamo (fecha, codigoEjemplar, tituloRecurso, nombreMaterial, codigoSocio)
values (to_char(sysdate,'dd/mm/yyyy'),1,'Star Wars Episode 3: Revenge of the Sith','Blu-ray',6);
```

```
insert into prestamo (fecha, codigoEjemplar, tituloRecurso, nombreMaterial, codigoSocio)
values ('05/06/2020',1,'Marca','Libro',1);
```

```
insert into pago (idPago, idPrestamo, Descripcion, tipoTransaccion, fecha, importeCuota)
values(1,2,'Deteriorado','Efectivo',to_char(sysdate,'dd/mm/yyyy'),0);
```

```
insert into pago (idPago, idPrestamo, Descripcion, tipoTransaccion, fecha, importeCuota)
values(1,3,'Correcto','Tarjeta',to_char(sysdate,'dd/mm/yyyy'),0);
```

```
insert into pago (idPago, idPrestamo, Descripcion, tipoTransaccion, fecha, importeCuota)
values(1,1,'Correcto','Efectivo',to_char(sysdate,'dd/mm/yyyy'),0);
```

```
insert into pago (idPago, idPrestamo, Descripcion, tipoTransaccion, fecha, importeCuota)
values(1,5,'Correcto','Tarjeta',to_char(sysdate,'dd/mm/yyyy'),0);
```

```
insert into pago (idPago, idPrestamo, Descripcion, tipoTransaccion, fecha, importeCuota)
values(1,4,'Correcto','Efectivo',to_char(sysdate,'dd/mm/yyyy'),0);
```

```
insert into pago (idPago, idPrestamo, Descripcion, tipoTransaccion, fecha, importeCuota)
values(1,7,'Correcto','Efectivo',to_char(sysdate,'dd/mm/yyyy'),0);
```

```
insert into Empleado values(1,'fsfsf43','Anakin Skywalker','123456','Mustafar','11111111A');
```

```
insert into Empleado values(2,'dgdg2434','Padme Amidala','234567','Coruscant','22222222B');
```

```
insert into Empleado values(3,'dkdo45','Satele Shan','345678','Alderaan','33333333C');
```

```
insert into Empleado values(4,'ańka6464','R2 D2','456789','Almacen Amazon','44444444D');
```

```
insert into Empleado values(5,'sdksc654','Asaaj Ventress','567890','Quesh','55555555E');
```

```
insert into Empleado values(6,'sdjsicj54','Jar Jar Binks','678901','Tatooine','66666666H');
```

```
insert into Empleado values(7,'pcvkd65','Nihilus Traya','789012','Malachor V','77777777I');
```

```
insert into Empleado values(8,'jvjcj797','C3 PO','890123','Naboo','88888888J');
```

```
insert into Empleado values(9,'jvjv64','Meetra Surik','901234','Dantooine','99999999K');
```

```
insert into Empleado values(10,'joxjv464','Rubi Malone','012345','Borde Exterior','00000000L');
```

```
insert into TrabajaEmpBiblio values('Uhu',1);
```

```
insert into TrabajaEmpBiblio values('Uhu',2);
```

```
insert into TrabajaEmpBiblio values('Uhu',3);
```

```
insert into TrabajaEmpBiblio values('Uma',4);
```



```
insert into TrabajaEmpBiblio values('Publica de Granada',5);
insert into TrabajaEmpBiblio values('Publica de Granada',6);
insert into TrabajaEmpBiblio values('Complutense',7);
insert into TrabajaEmpBiblio values('Ugr',8);
insert into TrabajaEmpBiblio values('Ugr',9);
insert into TrabajaEmpBiblio values('Ugr',10);
```

```
insert into RealizaAnotacionSocio values(1,3,to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionSocio values(1,7,to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionSocio values(4,8,to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionSocio values(8,8,to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionSocio values(6,2,to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionSocio values(3,1,to_char(sysdate,'dd/mm/yyyy'));
```

```
insert into RealizaAnotacionRecurso values(2,'Star Wars Episode 3: Revenge of the Sith','Blu-
ray',to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionRecurso values(5,'Prison Break','Blu-
ray',to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionRecurso values(6,'Harry Potter y el misterio del
principe','Libro',to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionRecurso values(3,'The Best of 2008-
2012','CD',to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionRecurso values(2,'Marca','Libro',to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionRecurso values(8,'La Senda
Jedi','Libro',to_char(sysdate,'dd/mm/yyyy'));
```

```
insert into RealizaAnotacionPago values(2,1,2,to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionPago values(6,1,1,to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionPago values(4,1,4,to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionPago values(4,1,3,to_char(sysdate,'dd/mm/yyyy'));
insert into RealizaAnotacionPago values(10,1,2,to_char(sysdate,'dd/mm/yyyy'));
```

7.3. Update

```
update HistorialEstado set codigoEjemplar=3 where idHistorial=1;
update HistorialEstado set codigoEjemplar=1 where idHistorial=2;
update HistorialEstado set codigoEjemplar=1 where idHistorial=3;
update HistorialEstado set codigoEjemplar=1 where idHistorial=4;
update HistorialEstado set codigoEjemplar=1 where idHistorial=5;
update HistorialEstado set codigoEjemplar=1 where idHistorial=6;
update HistorialEstado set codigoEjemplar=1 where idHistorial=7;
update HistorialEstado set codigoEjemplar=2 where idHistorial=8;
update HistorialEstado set codigoEjemplar=1 where idHistorial=9;
update HistorialEstado set codigoEjemplar=1 where idHistorial=10;
update HistorialEstado set codigoEjemplar=1 where idHistorial=11;
update HistorialEstado set codigoEjemplar=1 where idHistorial=12;
update HistorialEstado set codigoEjemplar=2 where idHistorial=13;
update HistorialEstado set codigoEjemplar=3 where idHistorial=14;
update HistorialEstado set codigoEjemplar=1 where idHistorial=15;
update HistorialEstado set codigoEjemplar=2 where idHistorial=16;
update HistorialEstado set codigoEjemplar=3 where idHistorial=17;
```

7.4. Delete

```
delete from prestamo where idPrestamo=7;
delete from pago where idPrestamo=7 and idPago=1;
```

8. OPERACIONES CON PL/SQL:

8.1. Cursores

Cursor 1 => Mostrar el importe total de las penalizaciones de cada socio.

```
set serveroutput on;
```

```
declare
```

```
    cursor c_totalPenalizacion is select prestamo.codigoSocio, socio.nombreApellido,
```

```
    sum(importePenalizacion) from pago, prestamo, socio where
```

```
pago.idPrestamo = prestamo.idPrestamo and prestamo.codigoSocio = socio.codigoSocio
group by prestamo.codigoSocio, socio.nombreApellido;
v_codigoSocio prestamo.codigoSocio%type;
v_nombreSocio socio.nombreApellido%type;
v_importePenalizacion pago.importePenalizacion%type;
begin
    open c_totalPenalizacion;
    loop
    fetch c_totalPenalizacion into v_codigoSocio, v_nombreSocio, v_importePenalizacion;
    exit when c_totalPenalizacion%notfound;
        dbms_output.put_line('El socio 'v_nombreSocio' tiene un total de
        'v_importePenalizacion'€ de penalización');
    end loop;
    close c_totalPenalizacion;
end;
```

Cursor 2 => Mostrar todos los títulos de blu-ray que han sido alquilados desde el inicio de las bibliotecas.

```
set serveroutput on;
declare
    cursor c_bluRay is select distinct ejemplar.tituloRecurso from ejemplar, historialEstado,
    estado where ejemplar.codigoEjemplar = historialEstado.codigoEjemplar and
    historialEstado.idHistorial = estado.idHistorial and ejemplar.nombreMaterial = 'Blu-ray' and
    estado.nombreEstado = 'alquilado';
    v_tituloRecurso ejemplar.tituloRecurso%type;
    v_cont int:=0;
    e_noAlquilados exception;
begin
    open c_bluRay;
    loop
        fetch c_bluRay into v_tituloRecurso;
```

```
        exit when c_bluRay%notfound;
        v_cont:=v_cont+1;
        dbms_output.put_line(v_tituloRecurso || ' blu-ray ha sido alquilado');
    end loop;
    close c_bluRay;
    if(v_cont=0) then
        raise e_noAlquilados;
    end if;

exception
    when e_noAlquilados then
        dbms_output.put_line('No se ha alquilado ningún blu-ray');
end;
```

Cursor 3 => Mostrar los libros pedidos de los últimos 6 meses

```
set serveroutput on;

declare

    cursor c_seisMeses is select ejemplar.codigoEjemplar, ejemplar.tituloRecurso,
        estado.nombreEstado from ejemplar, historialEstado, estado
    where ejemplar.codigoEjemplar = historialEstado.codigoEjemplar and ejemplar.tituloRecurso
        = historialEstado.tituloRecurso and
        ejemplar.nombreMaterial = historialEstado.nombreMaterial and
        historialEstado.idHistorial = estado.idHistorial
        and estado.nombreEstado = 'pedido' and ejemplar.nombreMaterial = 'Libro' and
        estado.fechaHora > (select add_months(trunc(sysdate,'MM'),-6) from dual);
    v_codigo ejemplar.codigoEjemplar%type;
    v_titulo ejemplar.tituloRecurso%type;
    v_estado estado.nombreEstado%type;
    v_cont int:=0;
    e_noLibrosPedidos exception;

begin
```

```
open c_seisMeses;
loop
    fetch c_seisMeses into v_codigo, v_titulo, v_estado;
    exit when c_seisMeses%notfound;
    v_cont:=v_cont+1;
    dbms_output.put_line(v_codigo' 'v_titulo| '| ' se ha pedido');
end loop;
close c_seisMeses;
if(v_cont=0) then
    raise e_noLibrosPedidos;
end if;

exception
    when e_noLibrosPedidos then
        dbms_output.put_line('No se han pedido libros en los último 6 meses');
end;
```

8.2. Triggers

-----Autoincrementos

-----Socio -----

```
create sequence secuencia_socio start with 1;
```

```
create or replace trigger tr_secuencia_socio before insert on Socio
```

```
for each row
```

```
begin
```

```
    select secuencia_socio.nextval into :new.codigoSocio from dual;
```

```
end;
```

----- HistorialEstado -----

```
create sequence secuencia_HistorialEstado start with 1;
```

```
create or replace trigger tr_secuencia_HistorialEstado before insert on HistorialEstado
```

```
for each row
begin
    select secuencia_HistorialEstado.nextval into :new.idHistorial from dual;
end;
```

----- Prestamo -----

```
create sequence secuencia_Prestamo start with 1;
```

```
create or replace trigger tr_secuencia_Prestamo before insert on Prestamo
```

```
for each row
```

```
begin
```

```
    select secuencia_Prestamo.nextval into :new.idPrestamo from dual;
```

```
end;
```

----- Empleado -----

```
create sequence secuencia_Empleado start with 1;
```

```
create or replace trigger tr_secuencia_Empleado before insert on Empleado
```

```
for each row
```

```
begin
```

```
    select secuencia_Empleado.nextval into :new.codigoEmp from dual;
```

```
end;
```

=> Trigger que calcula la media del tiempoLimite y numeroRenovacion de la tabla Prestamo, insertándolo en sus columnas correspondientes.

```
create or replace trigger tr_calculoPrestamoTiempos before insert on prestamo
```

```
for each row
```

```
declare
```

```
    v_limiteMaterial material.tiempoLimite%type;
```

```
    v_renovacionMaterial material.numeroRenovacion%type;
```

```
    v_limitePerfil perfil.tiempoLimite%type;
```

```
    v_renovacionPerfil perfil.numeroRenovacion%type;
```

```
v_mediat tiempoLimite numeric;
v_mediaNumeroRenovacion numeric;
v_limiteColeccion recurso.nombreColeccion%type;
begin
    select tiempoLimite into v_limiteMaterial from material where
        nombreMaterial=:new.nombreMaterial;
    select numeroRenovacion into v_renovacionMaterial from material where
        nombreMaterial=:new.nombreMaterial;
    select tiempoLimite into v_limitePerfil from perfil,socio where
        perfil.nombrePerfil=socio.nombrePerfil and codigoSocio=:new.codigoSocio;
    select numeroRenovacion into v_renovacionPerfil from perfil,socio where
        perfil.nombrePerfil=socio.nombrePerfil and codigoSocio=:new.codigoSocio;
    select tiempoLimite into v_limiteColeccion from coleccion,recurso where
        coleccion.nombreColeccion=recurso.nombreColeccion
        and recurso.tituloRecurso=:new.tituloRecurso and nombreMaterial=:new.nombreMaterial;

    --para tiempoLimite
    v_mediat tiempoLimite:=round((v_limiteMaterial+v_limitePerfil)/2);
    if(v_mediat tiempoLimite>v_limiteColeccion) then
        v_mediat tiempoLimite:=v_limiteColeccion;
    end if;

    --para numeroRenovacion
    v_mediaNumeroRenovacion:=round((v_renovacionMaterial+v_renovacionPerfil)/2);

    :new.tiempoLimite:=v_mediat tiempoLimite;
    :new.numeroRenovacion:=v_mediaNumeroRenovacion;
end;
```

=> Trigguer que tras realizarse un préstamo, añade el nuevo estado “alquilado”.

```
create or replace trigger tr_insertarEstadoAlquilado after insert on prestamo
for each row
declare
    v_idHistorial historialEstado.idHistorial%type;
begin
    select idHistorial into v_idHistorial from HistorialEstado where
        codigoEjemplar=:new.codigoEjemplar and tituloRecurso=:new.tituloRecurso
        and nombreMaterial=:new.nombreMaterial;

    insert into estado values(:new.fecha,v_idHistorial,'alquilado');
end;
```

=> Trigguer que calcula e inserta los días totales del préstamo.

```
create or replace trigger tr_diasPrestadosTotales before insert on pago
for each row
declare
    v_fechaInicial prestamo.fecha%type;
    v_penalizacionDias coleccion.penalizacionesEurosDia%type;
    v_penalizacionesActuales socio.penalizacionesActuales%type;
    v_codigoSocio socio.codigoSocio%type;
    v_tiempoLimite prestamo.tiempoLimite%type;
begin
    select fecha into v_fechaInicial from prestamo where idPrestamo= :new.idPrestamo;
    update prestamo set diasPrestadosTotales = diferenciaFechas(v_fechaInicial) where
        prestamo.idPrestamo=:new.idPrestamo;

    select penalizacionEurosDia, prestamo.tiempoLimite into v_penalizacionDias,
        v_tiempoLimite from coleccion, prestamo, recurso, ejemplar where
        prestamo.idPrestamo=:new.idPrestamo and prestamo.tituloRecurso =
        ejemplar.tituloRecurso and prestamo.nombreMaterial = Ejemplar.nombreMaterial and
        prestamo.codigoEjemplar = Ejemplar.codigoEjemplar and recurso.tituloRecurso =
```



```
ejemplar.tituloRecurso and recurso.nombreMaterial=Ejemplar.nombreMaterial and  
recurso.nombreColeccion=coleccion.nombreColeccion;
```

```
if (diferenciaFechas(v_fechaInicial)-v_tiempoLimite) > 0 then  
    :new.importePenalizacion:=v_penalizacionDias * (diferenciaFechas(v_fechaInicial)-  
    v_tiempoLimite);  
else  
    :new.importePenalizacion:=0;  
end if;
```

```
select penalizacionesActuales, Socio.codigoSocio into v_penalizacionesActuales,  
v_codigoSocio from Socio, Prestamo where prestamo.idPrestamo=:new.idPrestamo and  
prestamo.codigoSocio = socio.codigoSocio;  
update Socio set penalizacionesActuales = v_penalizacionesActuales+1 where  
Socio.codigoSocio = v_codigoSocio;  
end;
```

=> Trigger que impide realizar más préstamos en caso de haber alcanzado las penalizaciones máximas permitidas.

```
create or replace trigger tr_limiteMaximoPenalizaciones before insert on prestamo  
for each row  
declare  
    v_penalizaciones socio.penalizacionesActuales%type;  
    v_limitePenalizaciones perfil.limitePenalizaciones%type;  
begin  
    select penalizacionesActuales into v_penalizaciones from Socio where  
    socio.codigoSocio=:new.codigoSocio;  
    select limitePenalizaciones into v_limitePenalizaciones from perfil,socio where  
    socio.codigoSocio=:new.codigoSocio and  
    perfil.nombrePerfil = socio.nombrePerfil;
```

```
if(v_penalizaciones>=v_limitePenalizaciones)then
    raise_application_error(-20600,'Has superado el límite de penalizaciones. No puedes
    realizar más prestamos');
end if;
end;
```

=> Trigger que impide realizar un préstamo si el ejemplar se encuentra reservado.

```
create or replace trigger tr_noRenovacion before insert on prestamo
for each row
declare
    v_estado estado.nombreEstado%type;
begin
    select nombreEstado into v_estado from estado, historialEstado, ejemplar, prestamo where
    estado.idHistorial = historialEstado.idHistorial and historialEstado.codigoEjemplar =
    ejemplar.codigoEjemplar and historialEstado.tituloRecurso = ejemplar.tituloRecurso and
    historialEstado.nombreMaterial = ejemplar.nombreMaterial and ejemplar.codigoEjemplar =
    prestamo.codigoEjemplar and prestamo.tituloRecurso = ejemplar.tituloRecurso and
    prestamo.nombreMaterial = ejemplar.nombreMaterial and estado.nombreEstado =
    'reservado' and ejemplar.codigoEjemplar=:new.codigoEjemplar and
    ejemplar.tituloRecurso=:new.tituloRecurso and
    ejemplar.nombreMaterial=:new.nombreMaterial;
    if(v_estado='reservado')then
        raise_application_error(-20600,'Este artículo se encuentra reservado, no se puede
        alquilar.');
```

8.3. Procedimientos y funciones

Funcion 1 => Devuelve la diferencia entre la fecha introducida y la actual (utilizada en trigger).

```
create or replace function diferenciaFechas (fechainicial date)
return integer is
```

```
difference int;  
begin  
    select to_date(to_char(sysdate,'dd/mm/yyyy')) –  
        to_date(to_char(fechainicial,'dd/mm/yyyy')) into difference from dual;  
    return difference;  
end;
```

Funcion 2 => Muestre los ejemplares totales de un recurso dado su título y material.

```
create or replace procedure p_ejemplaresRecurso (titulo varchar2, material varchar2)  
is  
    totales number;  
begin  
    select count(codigoEjemplar) into totales from Ejemplar where tituloRecurso = titulo and  
        nombreMaterial = material group by tituloRecurso, nombreMaterial;  
    dbms_output.put_line('Hay 'totales' ejemplar(es) de 'titulo' de 'material);  
exception  
    when no_data_found then  
        dbms_output.put_line('No hay ningun ejemplar de 'titulo' de 'material);  
end;  
  
set serveroutput on;  
  
declare  
    titulo ejemplar.tituloRecurso%type:= '&introduce_el_titulo_del_ejemplar';  
    material ejemplar.nombreMaterial%type:= '&introduce_el_material_del_ejemplar';  
begin  
    p_ejemplaresRecurso (titulo, material);  
end;
```

Funcion 3 => Stock total de una biblioteca dada.

```
create or replace function f_stock(nombreBiblioteca varchar2)  
return integer is
```

```
    totales int;
    noExiste int;
begin
    select count(*) into noExiste from biblioteca where biblioteca.nombreBiblio =
    nombreBiblioteca;
    if noExiste = 0 then
        return -1;
    else
        select count(ejemplar.tituloRecurso) into totales from ejemplar, recurso, material,
        soporte, tieneBiblioSoporte, biblioteca where biblioteca.nombreBiblio = nombreBiblioteca
        and ejemplar.tituloRecurso = recurso.tituloRecurso and ejemplar.nombreMaterial =
        recurso.nombreMaterial and recurso.nombreMaterial = material.nombreMaterial
        and material.nombreSoporte = soporte.nombreSoporte and soporte.nombreSoporte =
        tieneBiblioSoporte.nombreSoporte and tieneBiblioSoporte.nombreBiblio =
        biblioteca.nombreBiblio;
        return totales;
    end if;
end;

set serveroutput on;
declare
    biblioteca varchar2(30):='&introduce_el_nombre_de_la_biblioteca';
begin
    if f_stock(biblioteca)=-1 then
        dbms_output.put_line('La biblioteca ' biblioteca ' no existe');
    else
        dbms_output.put_line('La biblioteca ' biblioteca ' tiene ' f_stock(biblioteca) ' de stock');
    end if;
end;
```

Function 4 => Obtener los días actuales del préstamo dado su código.

```
create or replace function f_diasPrestamo(codigoPrestamo numeric)
return integer is
    fechalnicio date;
    noExiste int;
begin
    select count(*) into noExiste from prestamo where prestamo.idPrestamo = codigoPrestamo;
    if noExiste = 0 then
        return -1;
    else
        select fecha into fechalnicio from prestamo where idPrestamo=codigoPrestamo;
        return diferenciaFechas(fechalnicio);
    end if;
end;

set serveroutput on;
declare
    codigoPrestamo numeric:='&Introduce_el_codigo_del_préstamo';
begin
    if f_diasPrestamo(codigoPrestamo) = -1 then
        dbms_output.put_line('El código del préstamo 'codigoPrestamo' no existe');
    else
        dbms_output.put_line('El código del préstamo 'codigoPrestamo' lleva
        'f_diasPrestamo(codigoPrestamo)' dias');
    end if;
end;
```

Funcion 5 => Renovación de préstamo

```
create or replace procedure p_renovar(codigoPrestamo numeric) is
    v_limiteMaterial material.tiempoLimite%type;
    v_limitePerfil perfil.tiempoLimite%type;
    v_limiteColeccion recurso.nombreColeccion%type;
```

```
v_mediat tiempoLimite numeric;
v_numRenovacion prestamo.numeroRenovacion%type;
v_limitePrestamo prestamo.tiempoLimite%type;
v_fechaInicial prestamo.fecha%type;

begin
    select material.tiempoLimite, perfil.tiempoLimite, coleccion.tiempoLimite,
    prestamo.numeroRenovacion, prestamo.tiempoLimite, prestamo.fecha into
    v_limiteMaterial, v_limitePerfil, v_limiteColeccion, v_numRenovacion, v_limitePrestamo,
    v_fechaInicial from prestamo, socio, perfil, ejemplar, recurso, material, coleccion where
    prestamo.codigoSocio = socio.codigoSocio and
    socio.nombrePerfil = perfil.nombrePerfil and prestamo.codigoEjemplar =
    ejemplar.codigoEjemplar and prestamo.tituloRecurso = ejemplar.tituloRecurso and
    prestamo.nombreMaterial = ejemplar.nombreMaterial and ejemplar.tituloRecurso =
    recurso.tituloRecurso and ejemplar.nombreMaterial = recurso.nombreMaterial and
    recurso.nombreMaterial = material.nombreMaterial and recurso.nombrecoleccion =
    coleccion.nombreColeccion and prestamo.idPrestamo=codigoPrestamo;

    if (v_numRenovacion>0) and (diferenciaFechas(v_fechaInicial) < v_limitePrestamo) then
        if (diferenciaFechas(v_fechaInicial)-v_limitePrestamo) <= 1 then
            v_mediat tiempoLimite:=round((v_limiteMaterial+v_limitePerfil)/2);
            if(v_mediat tiempoLimite>v_limiteColeccion) then
                v_mediat tiempoLimite:=v_limiteColeccion;
            end if;
            update prestamo set numeroRenovacion=v_numRenovacion-1 where
            idPrestamo=codigoPrestamo;
            update prestamo set tiempoLimite=v_limitePrestamo+v_mediat tiempoLimite where
            idPrestamo=codigoPrestamo;
        else
            dbms_output.put_line('Todavía no se puede renovar');
        end if;
    else

```

```
        dbms_output.put_line('No se pueden realizar más renovaciones');
    end if;
end;

set serveroutput on;

declare
    codPrestamo prestamo.idPrestamo%type:='&Introduce_el_código_del_préstamo';
begin
    p_renovar(codPrestamo);
end;
```