

Bases de Datos

Proyecto: Administración de Biblioteca

Grupo 01

Semestre: 2019 -2

Equipo 3:

• Chávez Troncoso Héctor Manuel

• Garciarebollo Rojas Diego Iñaki

• Juárez Munguía Brandon Jesús

• Salazar Aguilar Manuel

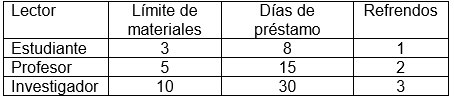
Fecha de entrega: 20 de mayo de 2019

* **Enunciado**

En una biblioteca se necesita llevar el control del préstamo de su material a sus lectores.

Todo material cuenta con un identificador, titulo, colocación, ubicación dentro de la biblioteca y autor; los autores están registrados con una clave de autor, nombre completo y nacionalidad. Se tienen dos tipos de materiales: libros y tesis; de los libros se tiene número de adquisición que es único, ISBN, tema y edición; las tesis están registradas con un identificador de tesis, carrera del tema de tesis, año de publicación, así como el director de tesis. Del director de tesis se tiene un identificador único, nombre completo y grado académico. De cada material se tiene por lo menos un ejemplar, para cada ejemplar se tiene número de ejemplar y estatus que indica si está disponible, en préstamo, no sale de la biblioteca o si esta en mantenimiento.

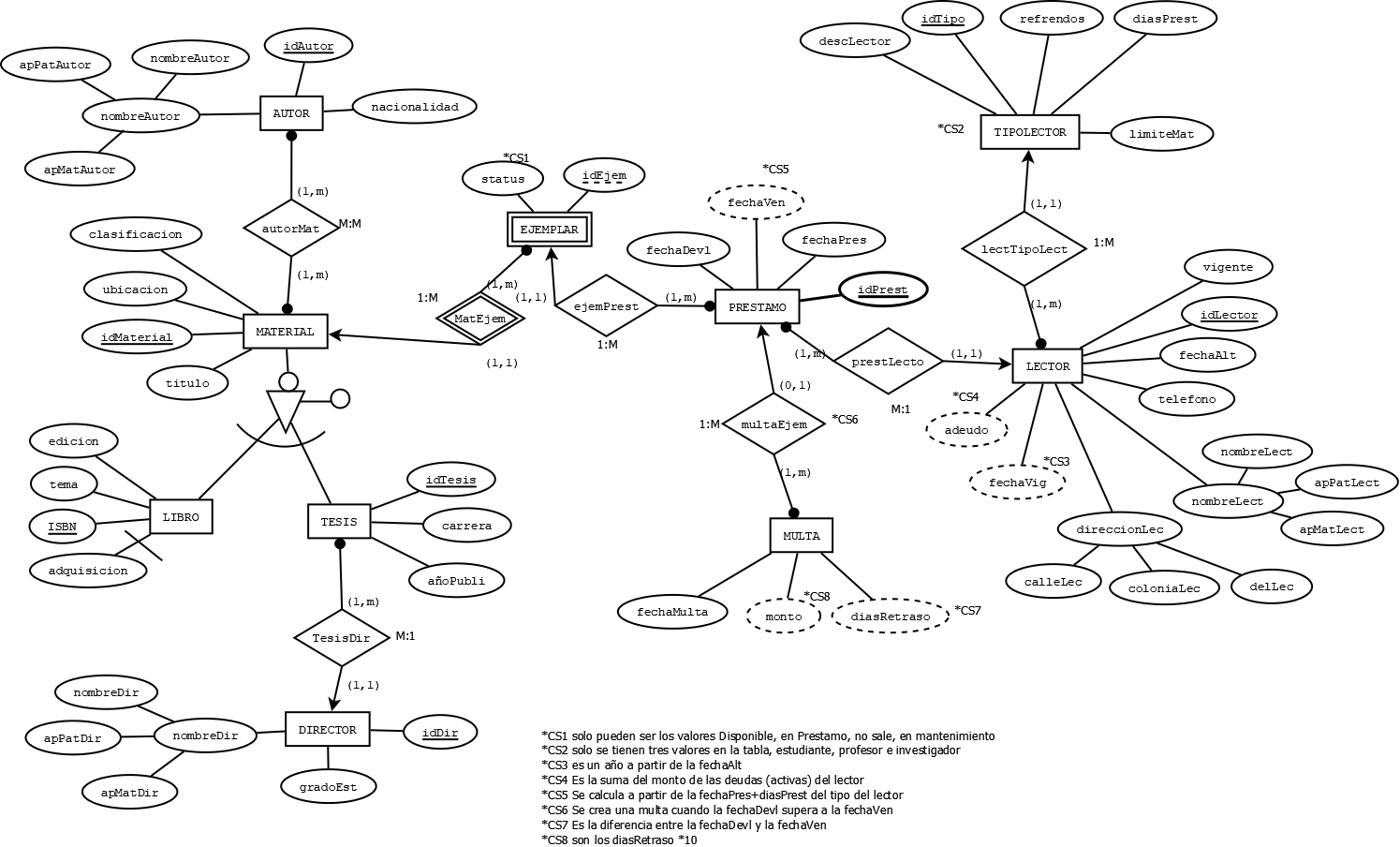
Los lectores están registrados por un identificador, tipo de lector (alumno, profesor, investigador), nombre completo, dirección, teléfono, adeudo, fecha en la cual se dio de alta en la biblioteca y fecha de vigencia (un año a partir de la fecha de alta en la biblioteca), dependiendo del tipo de lector es el límite de materiales que puede solicitar en préstamo, días permitidos, así como número de refrendos autorizados.



Al realizarse un préstamo es indispensable almacenar información sobre la fecha en la que se realiza, así como la fecha de vencimiento en base al tipo de lector; al realizarse una devolución es importante considerar que la fecha de devolución no exceda de la fecha de vencimiento o de lo contrario se registrará una multa (10 pesos por día de atraso) en la cual se debe de considerar el lector que lo solicito, el material en préstamo, la fecha de la multa, número de días de retraso y el monto al que se hizo acreedor, en el caso de resello se debe de considerar el número de refrendos permitidos de acuerdo al tipo de lector. Los refrendos se realizan únicamente en la fecha de vencimiento

**Diseño conceptual**

* **Modelo Entidad-Relación**



**Diseño Lógico**

* **Modelo relacional**
* **Diccionario de Datos**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| LIBRO | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idMaterial | CHAR(4) | FK | No | No | Si | PK, CASCADA | No |
| Isbn | CHAR(13) | U | No | No | Si | No | No |
| edicion | VARCHAR(20) | No | No | No | Si | No | No |
| tema | VARCHAR(20) | No | No | No | Si | No | No |
| adquisicion | CHAR(4) | U | No | No | Si | No | No |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TESIS | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idMaterial | CHAR(4) | FK | No | No | Si | PK, CASCADA | No |
| idTesis | CHAR(4) | U | No | No | Si | No | No |
| carrera | VARCHAR(30) | No | No | No | Si | No | No |
| añoPublicacion | DATE | No | No | No | Si | No | No |
| idDir | CHAR(4) | FK | No | No | No | FK,  EN NULO | No |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MATERIAL | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idMaterial | CHAR(4) | PK | No | No | Si | No | No |
| ubicacion | CHAR(18) | No | No | No | Si | No | No |
| clasificacion | CHAR(18) | No | No | No | Si | No | No |
| titulo | VARCHAR(30) | No | No | No | Si | No | No |
| tipoMaterial | CHAR(5) | No | LIBRO, TESIS | No | Si | CK | No |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| AUTORMAT | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idAutor | CHAR(4) | FK | No | No | Si | PK,  CASCADA | No |
| idMaterial | CHAR(4) | FK | No | No | Si | No |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| AUTOR | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idAutor | CHAR(4) | PK | No | No | Si | No | No |
| nacionalidad | VARCHAR(20) | No | No | No | Si | No | No |
| nombreAutor | VARCHAR(20) | No | No | No | Si | No | No |
| apPatAutor | VARCHAR(20) | No | No | No | Si | No | No |
| apMatAutor | VARCHAR(20) | No | No | No | No | No | No |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EJEMPLAR | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idMaterial | CHAR(4) | FK | No | No | Si | PK,  CASCADA | No |
| idEjem | CHAR(4) | D | No | No | Si | No |
| status | CHAR(13) | No | DISPONIBLE, PRESTAMO, NO SALE, MANTENIMIENTO | No | Si | CK | No |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DIRECTOR | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idDir | CHAR(4) | PK | No | No | Si | No | No |
| gradoEst | VARCHAR(30) | No | No | No | Si | No | No |
| nombreDir | VARCHAR(20) | No | No | No | Si | No | No |
| apPatDir | VARCHAR(20) | No | No | No | Si | No | No |
| apMatDir | VARCHAR(20) | No | No | No | No | No | No |

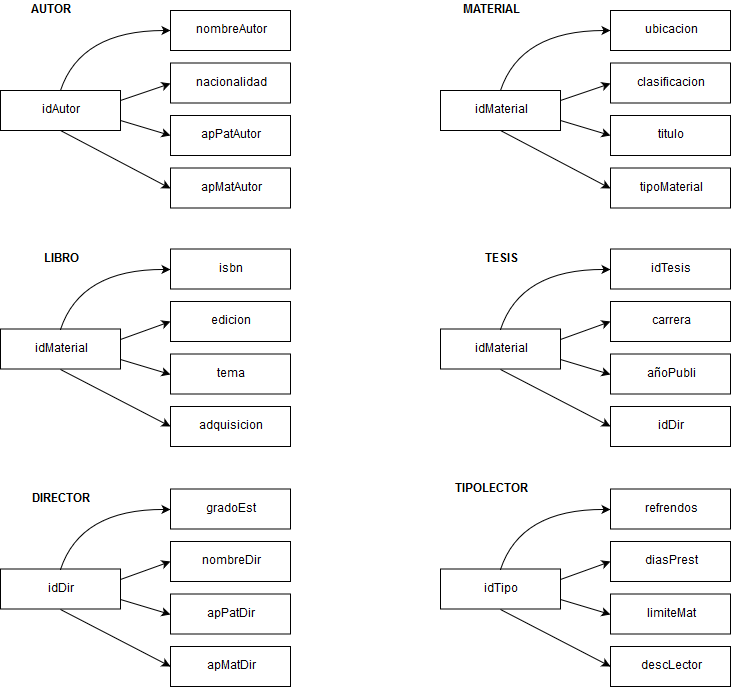
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PRESTAMO | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idPrestamo | CHAR(4) | PK | No | No | Si | No | No |
| fechaDevl | DATE | No | No | No | Si | No | No |
| fechaPres | DATE | No | No | SYSDATE | Si | No | No |
| idMaterial | CHAR(4) | FK | No | No | Si | FK,  CASCADE | No |
| idLector | CHAR(4) | FK | No | No | Si | No |
| idEjem | CHAR(4) | FK | No | No | Si | No |
| fechaVen | DATE | No | No | No | Si | No | Si |
| numRese | NUMBER(1) | No | No | 0 | Si | No | No |

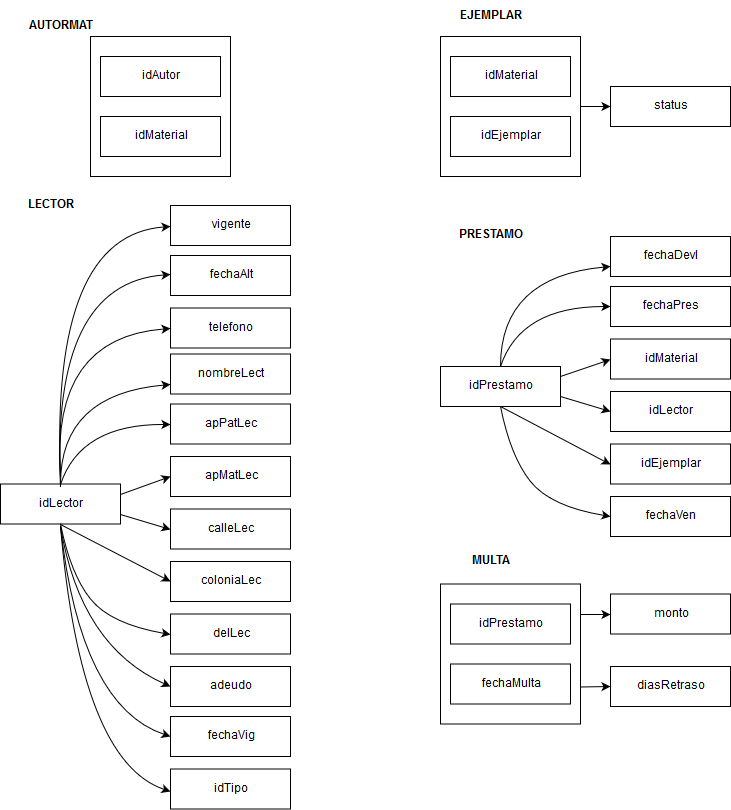
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| MULTA | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idPrestamo | CHAR(4) | FK | No | No | Si | PK,  CASCADE | No |
| fechaMulta | DATE | PK | No | SYSDATE | Si | No |
| monto | NUMBER(6) | No | No | No | Si | No | Si |
| diasRetraso | NUMBER(3) | No | No | No | Si | No | Si |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| LECTOR | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idLector | CHAR(4) | PK | No | No | Si | No | No |
| vigente | CHAR(1) | No | Y, N | No | Si | CK | No |
| fechaAlt | DATE | No | No | SYSDATE | Si | No | No |
| telefono | CHAR(10) | No | No | No | Si | No | No |
| nombreLect | VARCHAR(20) | No | No | No | Si | No | No |
| apPatLect | VARCHAR(20) | No | No | No | Si | No | No |
| apMatLect | VARCHAR(20) | No | No | No | No | No | No |
| calleLec | VARCHAR(30) | No | No | No | Si | No | No |
| coloniaLec | VARCHAR(50) | No | No | No | Si | No | No |
| delLec | VARCHAR(20) | No | No | No | Si | No | No |
| adeudo | NUMBER(5) | No | No | 0 | Si | No | Si |
| fechaVig | DATE | No | No | SYSDATE+365 | Si | No | No |
| idTipo | CHAR(4) | FK | No | No | Si | CASCADE | No |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TIPOLECTOR | | | | | | | |
| Nombre Atributo | **Tipo de Dato** | **Llave** | **Restricciones de dominio** | **Omisión** | **Obligatorio** | **Integridad** | **Derivado** |
| idTipo | CHAR(4) | PK | No | No | Si | No | No |
| referendos | NUMBER(1) | No | No | No | Si | No | No |
| diasPrest | NUMBER(1) | No | No | No | Si | No | No |
| limiteMat | NUMBER(1) | No | No | No | Si | No | No |
| descLector | VARCHAR(12) | No | No | No | Si | No | No |

* **Normalización**

****

****

**Diseño Físico**

* **DDL**

--@AUTORES: GARCIAREBOLLO

--JUAREZ SALAZAR

--@DESCRIPCION: SCRIPT ENCARGADO DE LA CREACION DE TABLAS DELMODELO RELACIONAL DEL PROYECTO BIBLIOTECA

--@FECHA:

CONNECT SYS AS SYSDBA;

CREATE USER adBiblio IDENTIFIED BY proyecto;

GRANT CONNECT, RESOURCE, CREATE VIEW, CREATE ANY INDEX TO adBiblio;

PROMPT INICIO DE SESION CON EL USUARIO CREADO PARA REALIZAR LA CREACION DE TABLAS adBiblio

CONNECT adBiblio;

--TABLA AUTOR

CREATE TABLE autor(

idAutor CHAR(4),

nacionalidad VARCHAR2(20) NOT NULL,

nombreAutor VARCHAR2(20) NOT NULL,

apPatAutor VARCHAR2(20) NOT NULL,

apMatAutor VARCHAR2(20) NULL,

CONSTRAINT PkAutor PRIMARY KEY(idAutor)

);

-- TABLA MATERIAL

CREATE TABLE material(

idMaterial CHAR(4),

ubicacion CHAR(18) NOT NULL,

clasificacion CHAR(18) NOT NULL,

titulo VARCHAR2(30) NOT NULL,

tipoMaterial CHAR(5) NOT NULL,

CONSTRAINT PkMaterial PRIMARY KEY (idMaterial),

CONSTRAINT CKTipoMaterial CHECK (tipoMaterial IN ('LIBRO','TESIS'))

);

-- TABLA AUTORMAT

CREATE TABLE autorMat(

idAutor CHAR(4),

idMaterial CHAR(4),

CONSTRAINT PkAutorMat PRIMARY KEY (idAutor,idMaterial),

CONSTRAINT FKAutorMatAut FOREIGN KEY (idAutor)

REFERENCES autor ON DELETE CASCADE,

CONSTRAINT FkAutorMatMat FOREIGN KEY (idMaterial)

REFERENCES material ON DELETE CASCADE

);

-- TABLA LIBRO

CREATE TABLE libro(

idMaterial CHAR(4),

isbn CHAR(13) NOT NULL,

edicion VARCHAR2(20) NOT NULL,

tema VARCHAR2(20) NOT NULL,

adquisicion CHAR(4) NOT NULL,

CONSTRAINT PkLibro PRIMARY KEY (idMaterial),

CONSTRAINT FklibroMat FOREIGN KEY (idMaterial)

REFERENCES material ON DELETE CASCADE,

CONSTRAINT UqLibroAdq UNIQUE (adquisicion),

CONSTRAINT UqLibroIsbn UNIQUE (isbn)

);

-- TABLA DIRECTOR

CREATE TABLE director(

idDir CHAR(4),

gradoEst VARCHAR2(30) NOT NULL,

nombreDir VARCHAR2(20) NOT NULL,

apPatDir VARCHAR2(20) NOT NULL,

apMatDir VARCHAR2(20),

CONSTRAINT PkDirector PRIMARY KEY (idDir)

);

-- TABLA TESIS

CREATE TABLE tesis(

idMaterial CHAR(4),

idTesis CHAR(4) NOT NULL,

carrera VARCHAR2(30) NOT NULL,

anioPubliCacion DATE NOT NULL,

idDir CHAR(4),

CONSTRAINT PkTesis PRIMARY KEY (idMaterial),

CONSTRAINT FKtesisMat FOREIGN KEY (idMaterial)

REFERENCES material ON DELETE CASCADE,

CONSTRAINT FktesisDir FOREIGN KEY (idDir)

REFERENCES director ON DELETE SET NULL,

CONSTRAINT UqTesis UNIQUE (idTesis)

);

-- TABLA EJEMPLAR

CREATE TABLE ejemplar(

idMaterial CHAR(4),

idEjem CHAR(4),

status CHAR(13) NOT NULL,

CONSTRAINT PkEjemplar PRIMARY KEY (idMaterial,idEjem),

CONSTRAINT FkEjemplarMat FOREIGN KEY (idMaterial)

REFERENCES material ON DELETE CASCADE,

CONSTRAINT CkEjemplarStat CHECK

(status IN('DISPONIBLE','PRESTAMO','MANTENIMIENTO','NO SALE'))

);

--- TABLA TIPOLECTOR

CREATE TABLE tipoLector(

idTipo CHAR(4),

refrendos NUMBER(1) NOT NULL,

diasPrest NUMBER(1) NOT NULL,

limiteMat NUMBER(1) NOT NULL,

descLector VARCHAR2(12) NOT NULL,

CONSTRAINT PkTipoLector PRIMARY KEY(idTipo)

);

-- TABLA LECTOR

CREATE TABLE lector(

idLector CHAR(4),

vigente CHAR(1) DEFAULT 'Y',

fechaAlta DATE DEFAULT SYSDATE,

telefono CHAR(10) NOT NULL,

nombreLect VARCHAR(20) NOT NULL,

apPatLect VARCHAR(20) NOT NULL,

apMatLect VARCHAR(20),

calleLect VARCHAR(30) NOT NULL,

coloniaLect VARCHAR(50) NOT NULL,

delLect VARCHAR(20) NOT NULL,

adeudo NUMBER(5) DEFAULT 0,

fechaVig DATE DEFAULT SYSDATE+365,

idTipo CHAR(4) NOT NULL,

CONSTRAINT PkLector PRIMARY KEY (idLector),

CONSTRAINT FkLectorTipo FOREIGN KEY (idTipo)

REFERENCES tipoLector ON DELETE CASCADE,

CONSTRAINT CkVig CHECK (vigente IN('Y','N'))

);

--- TABLA PRESTAMO

CREATE TABLE prestamo(

idPrestamo CHAR(4),

fechaDevl DATE NOT NULL,

fechaPres DATE DEFAULT SYSDATE,

idMaterial CHAR(4) NOT NULL,

idLector CHAR(4) NOT NULL,

idEjem CHAR(4) NOT NULL,

fechaVen DATE NOT NULL,

numRese NUMBER(1) DEFAULT 0,

hMulta CHAR(1) DEFAULT 'N',

CONSTRAINT PkPrestamo PRIMARY KEY (idPrestamo),

CONSTRAINT FkPrestamoEjem FOREIGN KEY (idMaterial,idEjem)

REFERENCES ejemplar ON DELETE CASCADE,

CONSTRAINT FkPrestamoLect FOREIGN KEY (idLector)

REFERENCES lector ON DELETE CASCADE

);

--- TABLA MULTA

CREATE TABLE multa(

idPrestamo CHAR(4),

fechaMulta DATE DEFAULT SYSDATE,

monto NUMBER(6) NOT NULL,

diasRetraso NUMBER(3) NOT NULL,

CONSTRAINT PkMulta PRIMARY KEY (idPrestamo,fechaMulta),

CONSTRAINT FkMultaPrest FOREIGN KEY(idPrestamo)

REFERENCES prestamo ON DELETE CASCADE

);

* **DML**

--- Se crea secuencia de la tabla autor

CREATE SEQUENCE seqAutor

MINVALUE 1

MAXVALUE 9999

START WITH 1

INCREMENT BY 1;

---------------------------------------------------------------------------

--- Se crea secuencia de la tabla prestamo

CREATE SEQUENCE seqPrestamo

MINVALUE 1

MAXVALUE 9999

START WITH 1

INCREMENT BY 1;

----------------------------------------------------------

--- Se crea secuencia de la tabla material

CREATE SEQUENCE seqMaterial

MINVALUE 1

MAXVALUE 9999

START WITH 1

INCREMENT BY 1;

----------------------------------------------------------

--- Se crea secuencia de la tabla adquisición

CREATE SEQUENCE seqAdquisicion

MINVALUE 1

MAXVALUE 9999

START WITH 1

INCREMENT BY 1;

---------------------------------------------------------------------------

--- Se crea secuencia de la tabla Director

CREATE SEQUENCE seqDir

MINVALUE 1

MAXVALUE 9999

START WITH 1

INCREMENT BY 1;

---------------------------------------------------------------------------

--- Se crea secuencia de la tabla Tesis

CREATE SEQUENCE seqTesis

MINVALUE 1

MAXVALUE 9999

START WITH 1

INCREMENT BY 1;

---------------------------------------------------------------------------

--- Se crea secuencia de la tabla Lector

CREATE SEQUENCE seqLector

MINVALUE 1

MAXVALUE 9999

START WITH 1

INCREMENT BY 1;

---10 INSERTS DE AUTORES--------

--- Datos pertenecientes a la tabla Autor

EXEC spInsertAutor(seqAutor.NEXTVAL,'mexicano','juan','rulfo');

EXEC spInsertAutor(seqAutor.NEXTVAL,'Ingles','J.R.R.','TOLKIEN');

EXEC spInsertAutor(seqAutor.NEXTVAL,'estadounidense','George R.R.','MARTIN');

EXEC spInsertAutor(seqAutor.NEXTVAL,'panameño','Carlos','Fuentes','Maciás');

EXEC spInsertAutor(seqAutor.NEXTVAL,'colombiano','Gabriel','garcia','marquez');

EXEC spInsertAutor(seqAutor.NEXTVAL,'estadounidense','Patrick','Rothfuss');

EXEC spInsertAutor(seqAutor.NEXTVAL,'mexicano','juan','perez','garcia');

EXEC spInsertAutor(seqAutor.NEXTVAL,'hondureño','gabriel','marquez','hernandez');

EXEC spInsertAutor(seqAutor.NEXTVAL,'frances','Albert','Camus');

EXEC spInsertAutor(seqAutor.NEXTVAL,'mexicano','gabriel','robles','sandoval');

---10 INSERTS DE LIBROS-----------------------------------------------------

--- Datos pertenecientes a la tabla Libro

EXEC spInsertLibro(seqMaterial.NEXTVAL,'Q5784A7890C447C789','NOVELA','Juego de Tronos','3','018457963571x','planeta','fantasia',seqAdquisicion.NEXTVAL);

EXEC spInsertLibro(seqMaterial.NEXTVAL,'c4785e4789f447f877','Infantil','El hobbit','2','784957610382a','minotauro','fantasia',seqAdquisicion.NEXTVAL);

EXEC spInsertLibro(seqMaterial.NEXTVAL,'E7845R1146F145W123','NOVELA','PEDRO PARAMO','1','147854693125C','debolsillo','realista',seqAdquisicion.NEXTVAL);

EXEC spInsertLibro(seqMaterial.NEXTVAL,'E7112E2156F154A122','Poemas','Coleccion de poemas','10','147854698212x','minotauro','literario',seqAdquisicion.NEXTVAL);

EXEC spInsertLibro(seqMaterial.NEXTVAL,'e1455f8441C123R555','TEXTO','COMO PROGRAMAR EN SQL','8','451752178888C','UNIVERSIDAD','consulta',seqAdquisicion.NEXTVAL);

EXEC spInsertLibro(seqMaterial.NEXTVAL,'F7774r4444e111f111','Novela','el extraño','9','147845293748x','editorial mundo','filosofia',seqAdquisicion.NEXTVAL);

EXEC spInsertLibro(seqMaterial.NEXTVAL,'F7774r4444e111f112','novela','La muerte de Artemio cruz','4','147885293748x','minotauro','realista',seqAdquisicion.NEXTVAL);

EXEC spInsertLibro(seqMaterial.NEXTVAL,'E7845R1177F145W123','Novela','amor en los tiempos del colera','5','157945854978x','debolsillo','fantasia',seqAdquisicion.NEXTVAL);

EXEC spInsertLibro(seqMaterial.NEXTVAL,'f7845R1146F145W123','Novela','el nombre del silencio','6','136948585412x','debolsillo','fantasia',seqAdquisicion.NEXTVAL);

EXEC spInsertLibro(seqMaterial.NEXTVAL,'E7895R1146F145W123','novela','los muertos no hablan','7','255567841453x','universal','policiaca',seqAdquisicion.NEXTVAL);

---10 INSERTS DE directores------------------------------------------------

--- Datos pertenecientes a la tabla director

EXEC SPINSERTDIR(seqDir.NEXTVAL,'universidad','juan','garcia','franco');

EXEC SPINSERTDIR(seqDir.NEXTVAL,'doctorado','diego','peres');

EXEC SPINSERTDIR(seqDir.NEXTVAL,'universidad','manuel','rosas','rojas');

EXEC SPINSERTDIR(seqDir.NEXTVAL,'maestria','brandon','gutierrez');

EXEC SPINSERTDIR(seqDir.NEXTVAL,'doctorado','gabriel','diaz');

EXEC SPINSERTDIR(seqDir.NEXTVAL,'posgrado','victor','soto','cabello');

EXEC SPINSERTDIR(seqDir.NEXTVAL,'doctorado','eduardo','villanueva','perez');

EXEC SPINSERTDIR(seqDir.NEXTVAL,'maestria','jose','castilla','sosa');

EXEC SPINSERTDIR(seqDir.NEXTVAL,'universidad','hector','campos');

EXEC SPINSERTDIR(seqDir.NEXTVAL,'maestria','irving','rosas','miranda');

---10 INSERTS DE tesis-----------------------------------------------------

--- Datos pertenecientes a la tabla tesis

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'E7895R2246F145W123','texto','tesis como pasar mario bros 1','9',seqTesis.NEXTVAL,'sistemas','07/05/1998','1');

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'a8895R2246F145W123','texto','mov. de una turbina eolica','8',seqTesis.NEXTVAL,'mecanica','01/02/2018','2');

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'E7895R2246F445W123','texto','capacitacion de sql en redes','10',seqTesis.NEXTVAL,'sistemas','01/02/2012','2');

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'E7895R2246F145c123','texto','teologia en el dia a dia','9',seqTesis.NEXTVAL,'literatura','13/05/2009','5');

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'E7895R2246F145p222','texto','software libre y su economia','9',seqTesis.NEXTVAL,'sistemas','31/05/2007','6');

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'E7895R2246F145r333','texto','trabajos de recuperacion','8',seqTesis.NEXTVAL,'arte','11/02/1999','8');

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'E7895R2246F147v888','texto','economia mundial y mexico','10',seqTesis.NEXTVAL,'economia','30/06/2005','9');

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'E7895R2245c555e222','texto','literatura mexicana','8',seqTesis.NEXTVAL,'literatura','28/02/2000','10');

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'E7895R2246w774c444','texto','muerte sociocultural en mex','8',seqTesis.NEXTVAL,'literatura','13/05/1985','4');

EXEC SPINSERTTESIS(seqMaterial.NEXTVAL,'E7895c5556F145W123','texto','documentos en linea y su uso','10',seqTesis.NEXTVAL,'sistemas','14/05/2006','5');

---3 INSERTS DE tipolector-------------------------------------------------

--- Datos pertenecientes a la tabla tipoLector

EXEC SPINSERTTIPOLEC('1',1,8,3,'estudiante');

EXEC SPINSERTTIPOLEC('2',2,15,5,'profesor');

EXEC SPINSERTTIPOLEC('3',3,30,10,'investigador');

---10 INSERTS DE Lectores--------------------------------------------------

--- Datos pertenecientes a la tabla Lector

EXEC spInsertLector(seqLector.NEXTVAL,'5587496813','azucenas','tezonco','iztapalapa','1','juan','reyes');

EXEC spInsertLector(seqLector.NEXTVAL,'5214789652','hortencias','roma','alvaro obregon','2','manuel','perez','perez');

EXEC spInsertLector(seqLector.NEXTVAL,'5547813549','alelies','tezonco','iztapalapa','2','federico','rojas','ugalde');

EXEC spInsertLector(seqLector.NEXTVAL,'5531469728','claveles','nopalera','benito juarez','1','jose','hernandez');

EXEC spInsertLector(seqLector.NEXTVAL,'5531487493','rio nilo','coapa','magdalena','3','enrique','prieto');

EXEC spInsertLector(seqLector.NEXTVAL,'5518413541','francia','zapotitian','milpalta','2','claudia','esquivel','ramirez');

EXEC spInsertLector(seqLector.NEXTVAL,'5516841358','el salvador','san cristobal','benito juarez','3','laura','gomez','pardo');

EXEC spInsertLector(seqLector.NEXTVAL,'5567481855','colon','coapa','alvaro obregon','1','diana','romero');

EXEC spInsertLector(seqLector.NEXTVAL,'5541681392','cristobal','polanco','magdalena','1','tamara','villanueva');

EXEC spInsertLector(seqLector.NEXTVAL,'5514813866','nicolas campion','roma','alvaro obregon','3','lupita','perez','hernandez');

---15 INSERTS DE ejemplares------------------------------------------------

--- Datos pertenecientes a la tabla ejemplares

EXEC spInsertEjemplar('3','1','disponible');

EXEC spInsertEjemplar('5','1','disponible');

EXEC spInsertEjemplar('12','1','disponible');

EXEC spInsertEjemplar('13','1','disponible');

EXEC spInsertEjemplar('8','1','disponible');

EXEC spInsertEjemplar('1','1','no sale');

EXEC spInsertEjemplar('19','1','disponible');

EXEC spInsertEjemplar('20','1','no sale');

EXEC spInsertEjemplar('4','1','mantenimiento');

EXEC spInsertEjemplar('1','2','disponible');

EXEC spInsertEjemplar('6','1','disponible');

EXEC spInsertEjemplar('7','1','disponible');

EXEC spInsertEjemplar('2','1','disponible');

EXEC spInsertEjemplar('4','2','disponible');

EXEC spInsertEjemplar('4','3','disponible');

---10 INSERTS DE prestamos----------------------------------------------------

--- Datos pertenecientes a la tabla prestamo

EXEC spPrestamo(seqPrestamo.NEXTVAL,'3','1','2');

EXEC spPrestamo(seqPrestamo.NEXTVAL,'5','1','3');

EXEC spPrestamo(seqPrestamo.NEXTVAL,'4','1','5');

EXEC spPrestamo(seqPrestamo.NEXTVAL,'4','3','4');

EXEC spPrestamo(seqPrestamo.NEXTVAL,'12','1','8');

EXEC spPrestamo(seqPrestamo.NEXTVAL,'13','1','1');

EXEC spPrestamo(seqPrestamo.NEXTVAL,'20','1','1');

EXEC spPrestamo(seqPrestamo.NEXTVAL,'19','1','6');

EXEC spPrestamo(seqPrestamo.NEXTVAL,'7','1','6');

EXEC spPrestamo(seqPrestamo.NEXTVAL,'8','1','8');

* **PL/SQL**
  + **Procedimientos**

--@AUTORES: GARCIAREBOLLO

--JUAREZ SALAZAR

--@DESCRIPCION: SCRIPT ENCARGADO DE LA CREACION DE PROCEDIMIENTOS DEL PROYECTO BIBLIOTECA

--@FECHA:

--- Procedimiento encargado de insertar los datos Autor

CREATE OR REPLACE PROCEDURE spInsertAutor(

vIdAutor autor.idAutor%TYPE,

vNacionalidad autor.nacionalidad%TYPE,

vNombre autor.nombreAutor%TYPE,

vApPat autor.apPatAutor%TYPE,

vApMat autor.apMatAutor%TYPE:=NULL

)

AS BEGIN

INSERT INTO autor

VALUES (UPPER(vIdAutor),UPPER(vNacionalidad),UPPER(vNombre),

UPPER(vApPat),UPPER(vApMat));

END spInsertAutor;

/

------------------------------------------------------------------------------

--- Procedimiento encargado de insertar los datos director de tesis

CREATE OR REPLACE PROCEDURE spInsertDir(

vIdDir director.IdDir%TYPE,

vGradoEst director.gradoEst%TYPE,

vnombre director.nombreDir%TYPE,

vapPat director.apPatDir%TYPE,

vapMat director.apMatDir%TYPE:=NULL

)

AS BEGIN

INSERT INTO director

VALUES (UPPER(vIdDir),UPPER(vGradoEst),UPPER(vNombre),

UPPER(vApPat),UPPER(vApMat));

END spInsertDir;

/

------------------------------------------------------------------------------

--- Procedimiento encargado de insertar los datos de la tesis

CREATE OR REPLACE PROCEDURE spInsertTesis(

vIdMaterial material.idMaterial%TYPE,

vUbicacion material.ubicacion%TYPE,

vClasificacion material.clasificacion%TYPE,

vTitulo material.titulo%TYPE,

vIdAutor autor.IdAutor%TYPE,

vIdTesis Tesis.IdTesis%TYPE,

vCarrera Tesis.Carrera%TYPE,

vAnio Tesis.AnioPublicacion%TYPE,

vIdDir Tesis.IdDir%TYPE

)

AS BEGIN

INSERT INTO material

VALUES (UPPER(vIdMaterial),UPPER(vUbicacion),

UPPER(vClasificacion),UPPER(vTitulo),'TESIS');

INSERT INTO tesis

VALUES (UPPER(vIdMaterial),UPPER(vIdTesis),UPPER(vCarrera),

UPPER(vanio),UPPER(vIdDir));

INSERT INTO autorMat

VALUES (UPPER(vIdAutor),UPPER(vIdMaterial));

END spInsertTesis;

/

------------------------------------------------------------------------------

--- Procedimiento encargado de insertar los datos del libro

CREATE OR REPLACE PROCEDURE spInsertLibro(

vIdMaterial material.idMaterial%TYPE,

vUbicacion material.ubicacion%TYPE,

vClasificacion material.clasificacion%TYPE,

vTitulo material.titulo%TYPE,

vtipoMaterial material.tipoMaterial%TYPE,

vIdAutor autor.IdAutor%TYPE,

vISBN libro.isbn%TYPE,

vEdicion libro.edicion%TYPE,

vTema libro.tema%TYPE,

vAdquisicion libro.adquisicion%TYPE

)

AS BEGIN

INSERT INTO material

VALUES (UPPER(vIdMaterial),UPPER(vUbicacion),

UPPER(vClasificacion),UPPER(vTitulo),'LIBRO');

INSERT INTO libro

VALUES(UPPER(vIdMaterial),UPPER(vIsbn),UPPER(vEdicion),

UPPER(vTema),UPPER(vAdquisicion));

INSERT INTO autorMat

VALUES (UPPER(vIdAutor),UPPER(vIdMaterial));

END spInsertLibro;

/

------------------------------------------------------------------------------

--- Procedimiento encargado de insertar los datos relacionados a un ejemplar

CREATE OR REPLACE PROCEDURE spInsertEjemplar(

vidMaterial ejemplar.idMaterial%TYPE,

vidEjem ejemplar.IdEjem%TYPE,

vStatus ejemplar.status%TYPE

)

AS BEGIN

INSERT INTO ejemplar

VALUES (UPPER(vIdMaterial),UPPER(vIdEjem),UPPER(vStatus));

END spInsertEjemplar;

/

------------------------------------------------------------------------------

--- Procedimiento encargado de insertar los datos en relación al tipo de lector

CREATE OR REPLACE PROCEDURE spInsertTipoLec(

vIdTipo TipoLector.IdTipo%TYPE,

vRefrendos TipoLector.Refrendos%TYPE,

vDiasPrest TipoLector.DiasPrest%TYPE,

vLimiteMat TipoLector.LimiteMat%TYPE,

vDescrip TipoLector.DescLector%TYPE

)

AS BEGIN

INSERT INTO TipoLector

VALUES(UPPER(vIdTipo),UPPER(vRefrendos),

UPPER(vDiasPrest),UPPER(vLimiteMat),UPPER(vDescrip));

END spInsertTipoLec;

/

--------------------------------------------------------------------------------- Procedimiento encargado de insertar los datos en relación al lector

CREATE OR REPLACE PROCEDURE spInsertLector(

vIdLector Lector.IdLector%TYPE,

vTelefono Lector.Telefono%TYPE,

vCalle Lector.CalleLect%TYPE,

vColonia Lector.ColoniaLect%TYPE,

vDel Lector.DelLect%TYPE,

vIdTipo Lector.IdTipo%TYPE,

vNombre Lector.NombreLect%TYPE,

vApPat Lector.ApPatLect%TYPE,

vApMat Lector.ApMatLect%TYPE:=NULL

)

AS BEGIN

INSERT INTO lector(IdLector,Telefono,NombreLect,ApPatLect,ApMatLect,

CalleLect,ColoniaLect,DelLect,IdTipo)

VALUES(UPPER(vIdLector),vTelefono,UPPER(vNombre),UPPER(vApPat),UPPER(vApMat),UPPER(vCalle),UPPER(vColonia),UPPER(vDel),UPPER(vIdTipo));

END spInsertLector;

/

--------------------------------------------------------------------------------- Procedimiento encargado de insertar los datos en relación a un prestamo

CREATE OR REPLACE PROCEDURE spPrestamo(

vIdPrestamo prestamo.idPrestamo%TYPE,

vIdMaterial ejemplar.idMaterial%TYPE,

vIdEjem ejemplar.idEjem%TYPE,

vIdLector lector.idLector%TYPE

)

AS

vTipoLector lector.idTipo%TYPE;

vDiasTipoLector tipoLector.diasPrest%TYPE;

BEGIN

--LOS TRIGGERS SE ENCARGARAN DE VERIFICAR:

--QUE NO HAYA MULTA PARA EL idLector

--QUE EL EJEMPLAR TENGA EL STATUS DE DISPONIBLE status

--QUE EL CONTEO DE LIBROS SEA CORRECTO PARA EL idLEctor

--SELECCION DEL TIPO DE LECTOR

SELECT idTipo INTO vTipoLector

FROM lector

WHERE idLector = vIdLector;

--SELECCION DEL NUMERO DE DIAS

SELECT diasPrest INTO vDiasTipoLector

FROM tipoLector

WHERE idTipo = vTipoLector;

--INSERCION DE DATOS

INSERT INTO prestamo

(idPrestamo,fechaDevl,idMaterial,idLector,idEjem,fechaVen)

VALUES(vIdPrestamo,SYSDATE+vIdLector,vIdMaterial,vIdLector,

vIdEjem,SYSDATE+vIdLector);

--ACTUALIZACION DE STATUS DEL MATERIAL

UPDATE ejemplar

SET status = 'PRESTAMO'

WHERE idMaterial = vIdMaterial

AND IdEjem = vIdEjem;

END spPrestamo;

/

--------------------------------------------------------------------------------- Procedimiento encargado de insertar los datos en relación a una devolución

CREATE OR REPLACE PROCEDURE spDevol(

vIdMaterial ejemplar.idMaterial%TYPE,

vIdEjem ejemplar.idEjem%TYPE,

vIdLector lector.idLector%TYPE

)

AS

CURSOR curDev IS

SELECT idPrestamo

FROM prestamo p

WHERE p.idLector =vIdLector

AND p.idEjem = vIdEjem

AND p.idMaterial = vIdMaterial;

vIdPrestamo prestamo.idPrestamo%TYPE;

BEGIN

--EL TRIGGER tgPrestamo SECCION DELETING SE ENCARGARA DE VERIFICAR:

--QUE NO HAYA MULTA PARA EL idLector Y DE SER

--NECESARIO LA CREARA

BEGIN

OPEN curDev;

FETCH curDev INTO vIdPrestamo;

CLOSE curDev;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20020,'EL PRESTAMO SOLICITADO NO EXISTE');

END;

--BORRADO DEL PRESTAMO

DELETE FROM prestamo

WHERE idPrestamo = vIdPrestamo;

--ACTUALIZACION DEL STATUS DEL MATERIAL

UPDATE ejemplar

SET status = 'DISPONIBLE'

WHERE idMaterial = vIdMaterial

AND IdEjem = vIdEjem;

END spDevol;

/

--------------------------------------------------------------------------------- Procedimiento encargado de insertar los datos en relación a un resello

CREATE OR REPLACE PROCEDURE spResello(

vIdMaterial ejemplar.idMaterial%TYPE,

vIdEjem ejemplar.idEjem%TYPE,

vIdLector lector.idLector%TYPE

)

AS

CURSOR curDev IS

SELECT idPrestamo

FROM prestamo p

WHERE p.idLector =vIdLector

AND p.idEjem = vIdEjem

AND p.idMaterial = vIdMaterial;

vIdPrestamo prestamo.idPrestamo%TYPE;

vTipoLector lector.idTipo%TYPE;

vDiasTipoLector tipoLector.diasPrest%TYPE;

BEGIN

--EL TRIGGER tgPrestamo SECCION UPDATING SE ENCARGARA DE VERIFICAR:

--QUE EL DIA DE RESELLO SEA VALIDO

--SI SE EXCEDE SOLICITA QUE SE HAGA LA DEVOLUCION

--SI ESTA CON TIEMPO DE SOBRA SOLO RECHAZA EL RESELLO

--TAMBIEN VERIFICA QUE NO SE TENGA EL LIMITE DE REFRENDOS

BEGIN

OPEN curDev;

FETCH curDev INTO vIdPrestamo;

CLOSE curDev;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20020,'EL PRESTAMO SOLICITADO NO EXISTE');

END;

--SELECCION DEL TIPO DE LECTOR

SELECT idTipo INTO vTipoLector

FROM lector

WHERE idLector = vIdLector;

--SELECCION DEL NUMERO DE DIAS DE PRESTAMOS PARA

--REALIZAR LA ACTUALIZACION DE FECHAS

SELECT diasPrest INTO vDiasTipoLector

FROM tipoLector

WHERE idTipo = vTipoLector;

--ACTUALIZACION DEL PRESTAMO

UPDATE prestamo

SET fechaPres =SYSDATE,

fechaDevl = SYSDATE + vDiasTipoLector,

numRese = numRese+1

WHERE idPrestamo = vIdPrestamo;

END spResello;

/

--------------------------------------------------------------------------------- Procedimiento encargado de insertar los datos en relación a una multa

CREATE OR REPLACE PROCEDURE spPagMulta(

vIdLector lector.idLector%TYPE,

vIdPrestamo prestamo.idPrestamo%TYPE,

vMonto multa.monto%TYPE

)

AS

BEGIN

--SE LIQUIDA LA MULTA ENTONCES EN FUNCION DEL ADEUDO SE PERMITIRA

--O NO EL BORRADO

UPDATE lector

SET adeudo = adeudo - vMonto

WHERE idLector = vIdLector;

--SE ELIMINA EL REGISTRO DE MULTA

DELETE FROM multa

WHERE idPrestamo = vIdPrestamo;

DBMS\_OUTPUT.PUT\_LINE('SE HA LIQUIDADO LA MULTA ASOCIADA AL PRESTAMO '||vIdPrestamo);

DELETE FROM prestamo

WHERE idPrestamo = vIdPrestamo;

END spPagMulta;

/

------------------------------------------------------------------------------

CONNECT system/oracle

SELECT object\_name

FROM dba\_objects

WHERE object\_type = 'PROCEDURE' AND OWNER='ADBIBLIO';

CONNECT adbiblio/proyecto

* + **Funciones**

--@AUTORES: GARCIAREBOLLO

-- JUAREZ SALAZAR

--@DESCRIPCION: SCRIPT ENCARGADO DE LA CREACION DE FUNCIONES DEL PROYECTO BIBLIOTECA

--@FECHA: 16/05/19

--FUNCION ENCARGADA DEL CALCULO DE MULTAS

PROMPT CREACION DE LA FUNCION DE CALCULO DE MULTAS

CREATE OR REPLACE FUNCTION ftMulta (

pFechaDev prestamo.fechaDevl%TYPE

)

RETURN NUMBER

IS

vMulta NUMBER(3,0):=0;

BEGIN

vMulta:=TRUNC(TO\_NUMBER(SYSDATE-pFechaDev));

vMulta:=vMulta\*10;

RETURN(vMulta);

END ftMulta;

/

* + **Disparadores**

--@AUTORES: GARCIAREBOLLO

-- JUAREZ SALAZAR

--@DESCRIPCION: SCRIPT ENCARGADO DE LA CREACION DE LOS DISPARADORES DEL PROYECTO BIBLIOTECA

--@FECHA: 16/05/19

--FUNCION ENCARGADA DEL CALCULO DE MULTAS

CREATE OR REPLACE TRIGGER tgPrestamo

BEFORE INSERT OR DELETE OR UPDATE OF fechaPres

ON prestamo

FOR EACH ROW

DECLARE

-- VARIABLES PARA INSERTING

vMonto multa.monto%TYPE;

vNumLibros NUMBER;

vLimite tipoLector.limiteMat%TYPE;

vStatus ejemplar.status%TYPE;

vTipoLector lector.idTipo%TYPE;

vDiasTipoLector tipoLector.diasPrest%TYPE;

-- VARIABLES PARA UPDATING

vDifFecha NUMBER;

vResLimit NUMBER;

-- VARIABLES PARA DELETING

vMultaMonto NUMBER;

vAdeudo NUMBER;

CURSOR curMul IS

SELECT SUM(m.monto)

FROM prestamo p

JOIN multa m

ON (p.idPrestamo=m.idPrestamo)

WHERE p.idLector=:NEW.idLector

GROUP BY p.idLector;

CURSOR curPres IS

SELECT COUNT(\*)

FROM prestamo p

WHERE p.idLector=:NEW.idLector;

BEGIN

CASE

--SI SE INSERTA UN NUEVO PRESTAMO HABRA QUE VERIFICAR EL STATUS

--DEL MATERIAL, QUE NO HAYA MULTA, QUE EL NUMERO DE PRESTAMOS

--SEA CORRECTO Y NO SE EXCEDA

WHEN INSERTING THEN

--VERIFICAMOS QUE EL MATERIAL ESTE DISPONIBLE

SELECT status INTO vStatus

FROM ejemplar

WHERE idMaterial = :NEW.idMaterial

AND idEjem = :NEW.idEjem;

IF vStatus != 'DISPONIBLE' THEN

RAISE\_APPLICATION\_ERROR(-20007,'EL MATERIAL '||:NEW.idEjem||' NO ESTA DISPONIBLE

PARA PRESTAMO');

END IF;

--VERIFICAMOS SI EL USUARIO TIENE UNA MULTA

--ESTO SE HACE CON AYUDA DEL CURSOR, QUE

--GUARDARA EL MONTO TOTAL DE MULTAS

BEGIN

OPEN curMul;

FETCH curMul INTO vMonto;

CLOSE curMul;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

vMonto := 0;

END;

IF vMonto > 0 THEN

RAISE\_APPLICATION\_ERROR(-20008,'NO SE PUEDE REALIZAR EL PRESTAMO, EL LECTOR '||:NEW.idLector||'

TIENE UNA MULTA DE '||vMonto);

END IF;

--VERIFICAMOS QUE EL NUMERO DE PRESTAMOS NO SE

--EXCEDA

--SELECCION DEL TIPO DE LECTOR

SELECT idTipo INTO vTipoLector

FROM lector

WHERE idLector = :NEW.idLector;

--SELECCION DEL NUMERO MAXIMO DE LIBROS QUE PUEDE PEDIR

SELECT limiteMat INTO vLimite

FROM tipoLector

WHERE idTipo = vTipoLector;

BEGIN

OPEN curPres;

FETCH curPres INTO vNumLibros;

CLOSE curPres;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

vNumLibros := 0;

END;

IF vNumLibros >= vLimite THEN

RAISE\_APPLICATION\_ERROR(-20009,'NO SE PUEDE REALIZAR EL PRESTAMO, EL LECTOR '||:NEW.idLector||'

NO PUEDE SOLICITAR MAS MATERIAL QUE '||vLimite);

END IF;

--CUANDO SE HACE UN UPDATE DENTRO DE PRESTAMO SIGNIFICA QUE SE HARA UN RESELLO

--DEBE VERIFICARSE QUE LA FECHA DEL RESELLO SEA EL DIA LIMITE DEL ANTIGUO PRESTAMO

--TAMBIEN QUE EL NUMERO MAXIMO DE RESELLOS AUN NO SE TENGA

WHEN UPDATING THEN

--VERIFICAMOS QUE LA FECHA ACTUAL SEA LA FECHA DE VENCIMIENTO

vDifFecha := TRUNC(TO\_NUMBER(SYSDATE-:OLD.fechaDevl));

IF vDifFecha < 0 THEN

RAISE\_APPLICATION\_ERROR(-20010,'NO SE PUEDE REALIZAR EL RESELLO,NO ES EL DIA DE ENTREGA

DE ESTE MATERIAL');

ELSIF vDifFecha > 0 THEN

RAISE\_APPLICATION\_ERROR(-20011,'NO SE PUEDE REALIZAR EL RESELLO,SE HA EXCEDIDO EL DIA DE ENTREGA

DE ESTE MATERIAL POR FAVOR REGISTRE LA DEVOLUCION');

END IF;

--VERIFICAMOS QUE NO SE TENGA YA EL LIMITE DE RESELLOS

SELECT idTipo INTO vTipoLector

FROM lector

WHERE idLector = :NEW.idLector;

SELECT refrendos INTO vResLimit

FROM tipoLector

WHERE idTipo = vTipoLector;

IF :OLD.numRese >= vResLimit THEN

RAISE\_APPLICATION\_ERROR(-20012,'NO SE PUEDE REALIZAR EL RESELLO, SE HA ALCANZADO

EL LIMITE');

END IF;

--CUANDO SE HACE UN DELETE DENTRO DE PRESTAMO SIGNIFICA QUE SE HA HECHO UNA DEVOLUCION

--DEBE DE REGISTRARSE UNA NUEVA MULTA EN CASO DE SER NECESARIO

WHEN DELETING THEN

--ESTE CASO CUBRE LA PRIMERA INSERCION DE LA MULTA PARA ESTE PRESTAMO

vDifFecha := TRUNC(TO\_NUMBER(SYSDATE-:OLD.fechaDevl));

SELECT adeudo INTO vAdeudo

FROM lector

WHERE idLector = :OLD.idLector;

IF vDifFecha > 0 AND

:OLD.hMulta = 'N' THEN

DBMS\_OUTPUT.PUT\_LINE('EL MATERIAL SE HA DEVUELTO CON ATRASO

SE REGISTRARA UNA MULTA PARA EL USUARIO SOBRE EL PRESTAMO '||

:OLD.idPrestamo);

vMultaMonto := ftMulta(:OLD.fechaDevl);

INSERT INTO multa

(idPrestamo,monto,diasRetraso)

VALUES(:OLD.idPrestamo,vMultaMonto,vDifFecha);

--UNA BANDERA QUE NOS DICE QUE EL PRESTAMO YA TIENE UNA MULTA

UPDATE prestamo

SET hMulta = 'Y'

WHERE (idPrestamo = :OLD.idPrestamo);

RAISE\_APPLICATION\_ERROR(-20013,'SOLO SE PODRA BORRAR DE PRESTAMO SI

SE LIQUIDA LA MULTA DE '||vMultaMonto||' PESOS');

--ESTE CASO NO PERMITIRA EL BORRADO POR QUE YA SE TIENE REGISTRADA LA MULTA

--Y ESTA NO SE HA LIQUIDADO, SI EL ADEUDO ES 0 NO SE ENTRA AQUI,

--SE PERMITE EL BORRADO

ELSIF :OLD.hMulta = 'Y' AND vAdeudo > 0 THEN

RAISE\_APPLICATION\_ERROR(-20013,'SOLO SE PODRA BORRAR DE PRESTAMO SI

SE LIQUIDA LA MULTA DE '||vMultaMonto||' PESOS');

END IF;

END CASE;

END tgPrestamo;

/

* + **Vistas**

CREATE OR REPLACE VIEW vwlibros

AS

select idMAterial,TITULO,IDEJEM,STATUS

FROM material

NATURAL JOIN ejemplar

where tipoMAterial LIKE 'LIBRO';

------------------------------------------

CREATE OR REPLACE VIEW vwtesis

AS

select idMAterial,TITULO,IDEJEM,STATUS

FROM material

NATURAL JOIN ejemplar

where tipoMAterial LIKE 'TESIS';

---------------------------------------

CREATE OR REPLACE VIEW vwautores

AS

SELECT \* FROM AUTOR;

------------------------------------------------------------

CREATE OR REPLACE VIEW VWLECTOR

AS

SELECT IDTIPO,NOMBRELECT,APPATLECT,APMATLECT,ADEUDO,FECHAVIG

FROM LECTOR

ORDER BY IDTIPO;