



## AD-5. Diseño base de datos y PL/SQL (funciones-procedimientos)

BASE DE DATOS

MICHAEL TELLERIA GUADALAJARA  
PRIMERO DE DAM

## Enunciado

Debemos diseñar y crear, una base de datos para la gestión de prestamos en una biblioteca municipal. Los usuarios están registrados en la base de datos, y los libros están dados de alta con todos sus datos identificativos, ISBN, titulo, autor, etc.

Lo importante es el control de prestamos de libros que se hacen a los usuarios, la fecha de entrega y devolución, los días de penalización por falta de entrega a tiempo, enviar mensajes recordando la entrega y la penalización superior a 7 días que hace que usuario no pueda realizar prestamos hasta pasados 2 meses desde la devolución del libro con retraso en la entrega.

Debemos crear un procedimiento para insertar usuarios, con los datos necesarios para el alta de los mismos como parámetros. Otro procedimiento para dar de alta los libros según llegan a la biblioteca, con todos los datos que debemos informar en la tabla correspondiente. Y por último un procedimiento para insertar los prestamos en su correspondiente tabla.

Crearemos dos funciones para gestionar la información de prestamos:

**SANCION\_USUARIO:** Calculará la diferencia entre la fecha de préstamo y la del momento de cálculo, y devolverá una cadena que puede ser ACTIVO, GRAVE Y MUY GRAVE. Si los días devueltos son menores o iguales a 7, el préstamo será ACTIVO (no hay problema), si son mayores de 7 y menores de 12 el estado del préstamo será GRAVE y si es mayor de 12 será MUY GRAVE.

**MENSAJE\_SANCION:** Compondrá el siguiente mensaje como salida de la función (Estimado usuario <nombre y apellidos>, el pasado día <día préstamo> realizó un préstamo en nuestra biblioteca con código <código préstamo> y ha superado la fecha de entrega prevista en <días superación fecha entrega normal> días, por lo que le corresponde una sanción <GRAVE o MUY GRAVE>)

Por último, crearemos un procedimiento que llamaremos GESTOR, que recorrerá toda la tabla que recoge los prestamos y aplicará las funciones SANCION\_USUARIO y MENSAJE\_SANCION, e irá actualizando los campos NIVEL\_SANCION y MENSAJE, de esa misma tabla además de insertar en una tabla adicional llamada SANCIONADOS, los datos de CODIGO\_USUARIO,

FECHA\_PRESTAMO, DIAS\_SANCION y MENSAJE, de todos los usuarios que tengan un NIVEL\_SANCION, que sea GRAVE O MUY GRAVE.

Evaluación de la actividad

Se evaluará el diseño del diagrama E/R, DIAGRAMA FUNCIONAL, TABLAS, 2 funciones y 4 procedimientos.

### **Resolución de la actividad**

En primera instancia tras leer el enunciado, podemos identificar las entidades

USUARIO, LIBRO, PRESTAMO. (SE CREO UNA ENTIDAD LLAMADA SANCIONADOS PARA ALMACENAR LAS SANCIONES EN EL DIAGRAMA FUNCIONAL, PERO QUERIA QUE SE VEA REFLEJADO EN EL DIAGRAMA E/R )

Y realizar por primera vez el modelo E/R sin normalización.

### **Primera de forma de normalización(1FN).**

En la primera forma eliminamos los grupos repetitivos.

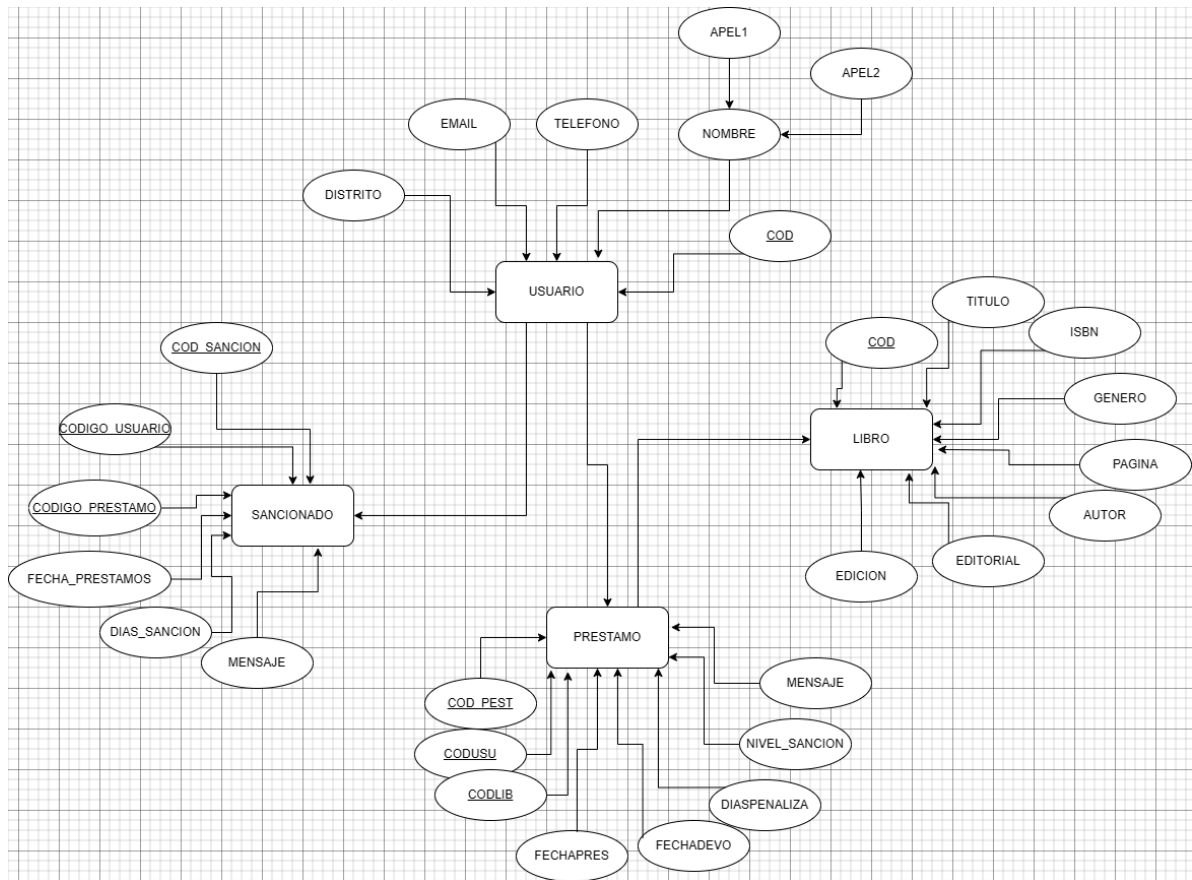
### **Segunda forma de normalización(2FN).**

Eliminamos dependencias parciales y creamos tablas separadas para conjuntos de datos que se aplican a múltiples registros.

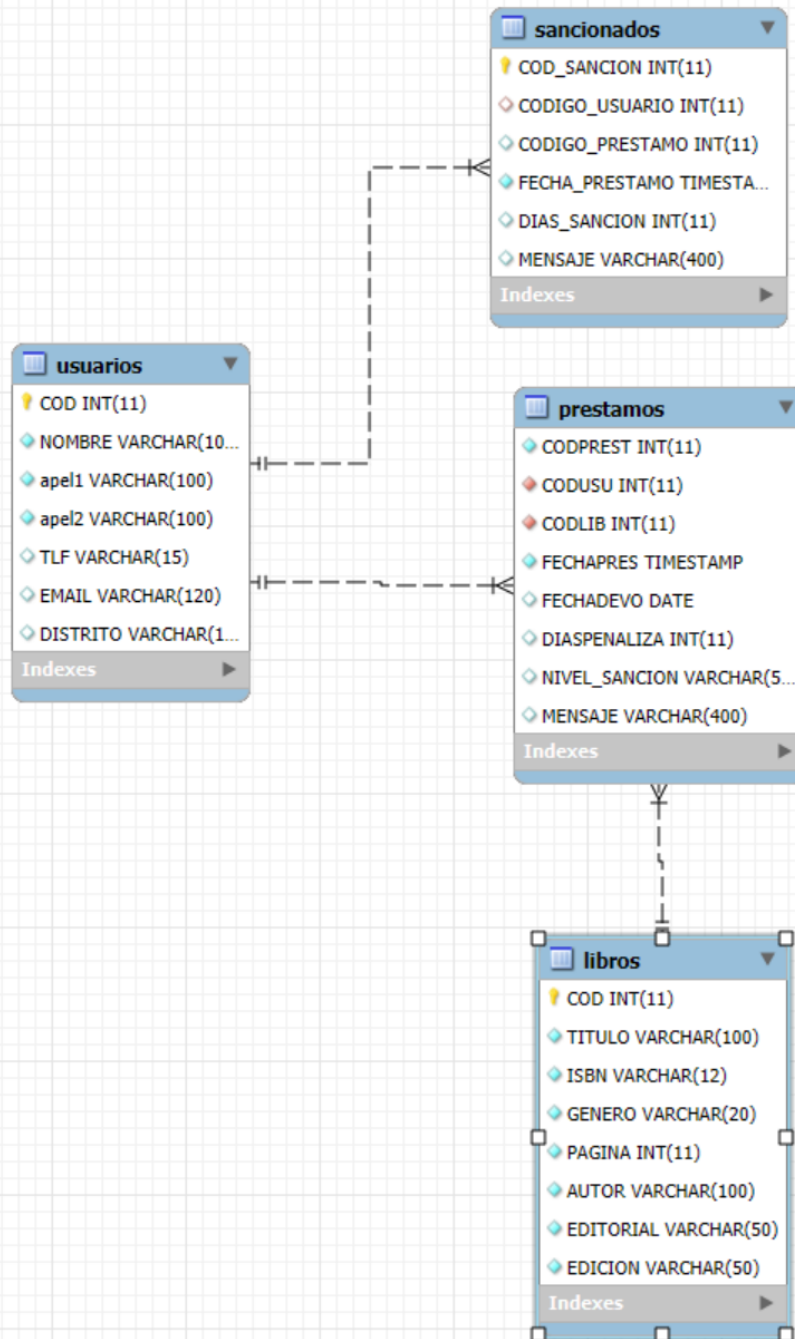
### **Tercera Forma normal(3FN).**

Eliminamos dependencias transitivas (en este caso, no era necesario), Creamos tablas separadas para conjuntos de campos que no dependan directamente de clave primaria.

## Modelo E/R



## Modelo Funcional



## Líneas de código de la creación de la anterior tabla

```

1
2  /* SELECCIONAMOS LA BASE DE DATOS*/
3  CREATE DATABASE BIBLIOTECA;
4  use biblioteca;
5
6  /* CREACION DE TABLAS*/
7  CREATE TABLE `libros` (
8      `COD` int(11) NOT NULL AUTO_INCREMENT,
9      `TITULO` varchar(100) NOT NULL,
10     `ISBN` varchar(12) NOT NULL,
11     `GENERO` varchar(20) NOT NULL,
12     `PAGINA` int NOT NULL,
13     `AUTOR` VARCHAR(100) NOT NULL,
14     `EDITORIAL` VARCHAR(50) NOT NULL,
15     `EDICION` VARCHAR(50) NOT NULL,
16     PRIMARY KEY (`COD`),
17     UNIQUE KEY `ISBN` (`ISBN`),
18     KEY `GENERO` (`GENERO`)
19 ) ;
20
21 CREATE TABLE `usuarios` (
22     `COD` int(11) NOT NULL AUTO_INCREMENT,
23     `NOMBRE` varchar(100) NOT NULL,
24     `apell` varchar(100) NOT NULL,
25     `apel2` varchar(100) NOT NULL,
26     `TLF` varchar(15) DEFAULT NULL,
27     `EMAIL` varchar(120) DEFAULT NULL,
28     `DISTRITO` varchar(15) DEFAULT NULL,
29     PRIMARY KEY (`COD`),
30     UNIQUE KEY `TLF` (`TLF`),
31     KEY `DISTRITO` (`DISTRITO`)
32 ) ;
33
34 CREATE TABLE `prestamos` (
35     `CODPREST` int(11) NOT NULL,
36     `CODUSU` int(11) NOT NULL,
37     `CODLIB` int(11) NOT NULL,
38     `FECHAPRES` timestamp NOT NULL DEFAULT current_timestamp(),
39     `FECHADEVO` date DEFAULT NULL,
40     `DIASPENALIZA` int(11) DEFAULT NULL,
41     `NIVEL_SANCIÓN` VARCHAR(50) DEFAULT NULL,
42     `MENSAJE` varchar(400) DEFAULT NULL,
43     KEY `CODUSU` (`CODUSU`),
44     KEY `CODLIB` (`CODLIB`),
45     CONSTRAINT `prestamos_ibfk_1` FOREIGN KEY (`CODUSU`) REFERENCES `usuarios` (`COD`) ON DELETE CASCADE ON UPDATE CASCADE,
46     CONSTRAINT `prestamos_ibfk_2` FOREIGN KEY (`CODLIB`) REFERENCES `libros` (`COD`) ON DELETE CASCADE ON UPDATE CASCADE
47 ) ;
48

```

## PROCEDIMIENTO

```

59  /* CREACION DE LAS STORED PROCEDURES*/
60
61  /* PRIMER PROCEDIMIENTO */
62
63  CREATE DEFINER='root'@'localhost' PROCEDURE `prest`(codpres int,codusu int,codlib int,fpres timestamp,fdevo date,diapen int,NIVEL_SANCIÓN VARCHAR(50),MENSAJE varchar(400))
64  BEGIN
65
66      insert into prestamos (CODPREST,CODUSU,CODLIB,FECHAPRES,FECHADEVO,DIASPENALIZA,NIVEL_SANCIÓN,MENSAJE)
67      values (codpres,codusu,codlib,fpres,fdevo,diapen,NIVEL_SANCIÓN,MENSAJE)
68
69  END
70

```

```

70 /* SEGUNDO PROCEDIMIENTO */
71
72
73 CREATE DEFINER='root'@'localhost' PROCEDURE `insertusu`(cod int,nomb varchar(50),apell varchar(50),apel2 varchar(50),tlf varchar(50),email varchar(100),dist varchar(50))
74 BEGIN
75
76     insert into usuarios (cod ,nombre,apell,apel2,tlf ,email,distrito)
77     values (cod ,nomb,apell,apel2,tlf ,email,dist);
78
79 END;
80
81 /* TERCER PROCEDIMIENTO */
82 CREATE DEFINER='root'@'localhost' PROCEDURE `altaLib`(cod int,titu varchar(50),isbn varchar(50),genr varchar(50),PAGINA int,AUTOR VARCHAR(100),EDITORIAL VARCHAR(50),EDICION VARCHAR(50))
83 BEGIN
84
85     insert into LIBROS (cod,titulo,isbn,genero,pag,AUTOR,EDITORIAL,EDICION)
86     values (cod,titu,isbn,genr,PAGINA,AUTOR,EDITORIAL,EDICION);
87
88 END;
89
90 /* CUARTO PROCEDIMIENTO*/
91
92 CREATE DEFINER='root'@'localhost' PROCEDURE `GESTOR` ()
93 BEGIN
94
95     UPDATE PRESTAMOS p
96     JOIN USUARIOS u ON p.CODUSU = u.COD
97     SET
98         p.NIVEL_SANCCION = SANCCION_USUARIO(p.FECHAPRES, p.FECHADEV),
99         p.MENSAJE = MENSAJES_SANCCION(u.NOMBRE, u.APELL1, u.APELL2, p.FECHAPRES, p.FECHADEV, p.CODPREST);
100
101
102
103 INSERT INTO SANCIONADOS (CODIGO_USUARIO, FECHA_PRESTAMO, DIAS_SANCCION, MENSAJE)
104
105 SELECT CODUSU,FECHAPRES,DATEDIFF(FECHADEV, FECHAPRES) AS DIAS_SANCCION,MENSAJE
106 FROM PRESTAMOS
107 WHERE NIVEL_SANCCION IN ('GRAVE', 'MUY GRAVE');
108
109
110 END;
111

```

## PRIMERA FUNCION

```

111 /* PRIMERA FUNCION*/
112
113
114 CREATE DEFINER='root'@'localhost' FUNCTION `SANCCION_USUARIO`(fpres timestamp,fdevo date) RETURNS varchar(50) CHARSET utf8mb4 COLLATE utf8mb4_general_ci
115 BEGIN
116 DECLARE resultado VARCHAR(50);
117 DECLARE dias INT;
118 SET dias = DATEDIFF(fdevo, fpres);
119
120 IF (dias < 7) THEN
121     SET resultado = 'ACTIVO';
122 ELSEIF (dias >= 7) AND (dias < 12) THEN
123     SET resultado = 'GRAVE';
124 ELSEIF (dias >= 12) THEN
125     SET resultado = 'MUY GRAVE';
126 END IF;
127
128 RETURN resultado;
129 END;

```

## SEGUNDA FUNCION

```

131 /* SEGUNDA FUNCION*/
132
133 CREATE DEFINER='root'@'localhost' FUNCTION `MENSAJES_SANCCION`(nomb varchar(50),APELL1 VARCHAR (50),APELL2 VARCHAR(50),FPREST timestamp,FDEVO DATE,CODPREST INT) RETURNS varchar(400) CHARSET utf8mb4 COLLATE utf8mb4_general_ci
134 BEGIN
135 DECLARE mensaje varchar (400);
136 DECLARE DIAS_RETRASO int;
137 DECLARE FDEVO_PREVISTA INT;
138 DECLARE RESULTADO varchar(50);
139
140
141 SET resultado = SANCCION_USUARIO(fprest, fdevo);
142 SET DIAS_RETRASO = DATEDIFF(FDEVO, FPREST);
143 SET FDEVO_PREVISTA = ?;
144
145
146 SET mensajes concat('Estimado usuario',' ',NOMB,' ',APELL1,' ',APELL2,' ',
147 'el pasado día',' ',DATE_FORMAT(FPREST, "dY/m/kd"),',' ','realizó un préstamo en nuestra biblioteca con código',
148 ' ',CODPREST,' ', 'Han pasado ', DIAS_RETRASO , 'días desde el préstamo','y ha superado la fecha de entrega prevista en '
149 , ' ',FDEVO_PREVISTA , ' ', 'días,por lo que le corresponde una sanción',' ',RESULTADO);
150
151
152
153 RETURN mensajes;
154 END;
155

```

## DATOS DE PRUEBA

```
155
156  /* DATOS DE PRUEBA */
157
158
159  * select * from usuarios;
160  * select * from libros;
161  * select * from prestamos;
162  * select * from sancionados;
163
164  * call insertusu(123,'pop','eye','pop','78787878','pop@gmail.com','san fransisco');
165  * call insertusu(234,'pap','can','pot','73244378','pap@gmail.com','Los Angeles');
166  * call insertusu(3456,'pep','pep','pot','74234878','pep@gmail.com','Detroid');
167
168  * call altaLib(234,'mil besos','94357469357','romace',445, 'JUAN', 'JUN', '8');
169  * call altaLib(234,'el principe','94357469357','AAAA',445, 'JUANSI', 'JUNU', '6');
170  * call prest(1,123,234,'1978-08-11','2025-04-11',0,'null','null');
171  * call prest(2,234,234,'1978-08-11','2025-04-11',0,'null','null');
172  * call prest(3,3456,234,'1978-08-11','2025-04-11',0,'null','null');
173
174  * select biblioteca.MENSAJES_SANCION('pop', 'eye', 'pop', '2025-01-9', '2025-04-11', 1);
175
176  * INSERT INTO SANCIONADOS (COD_SANCION, CODIGO_USUARIO, CODIGO_PRESTAMO, FECHA_PRESTAMO, DIAS_SANCION, MENSAJE)
177  VALUES (0, 123,1, '2025-01-09','0','N');
178
179  * call biblioteca.GESTOR();
180
181
182
183
```