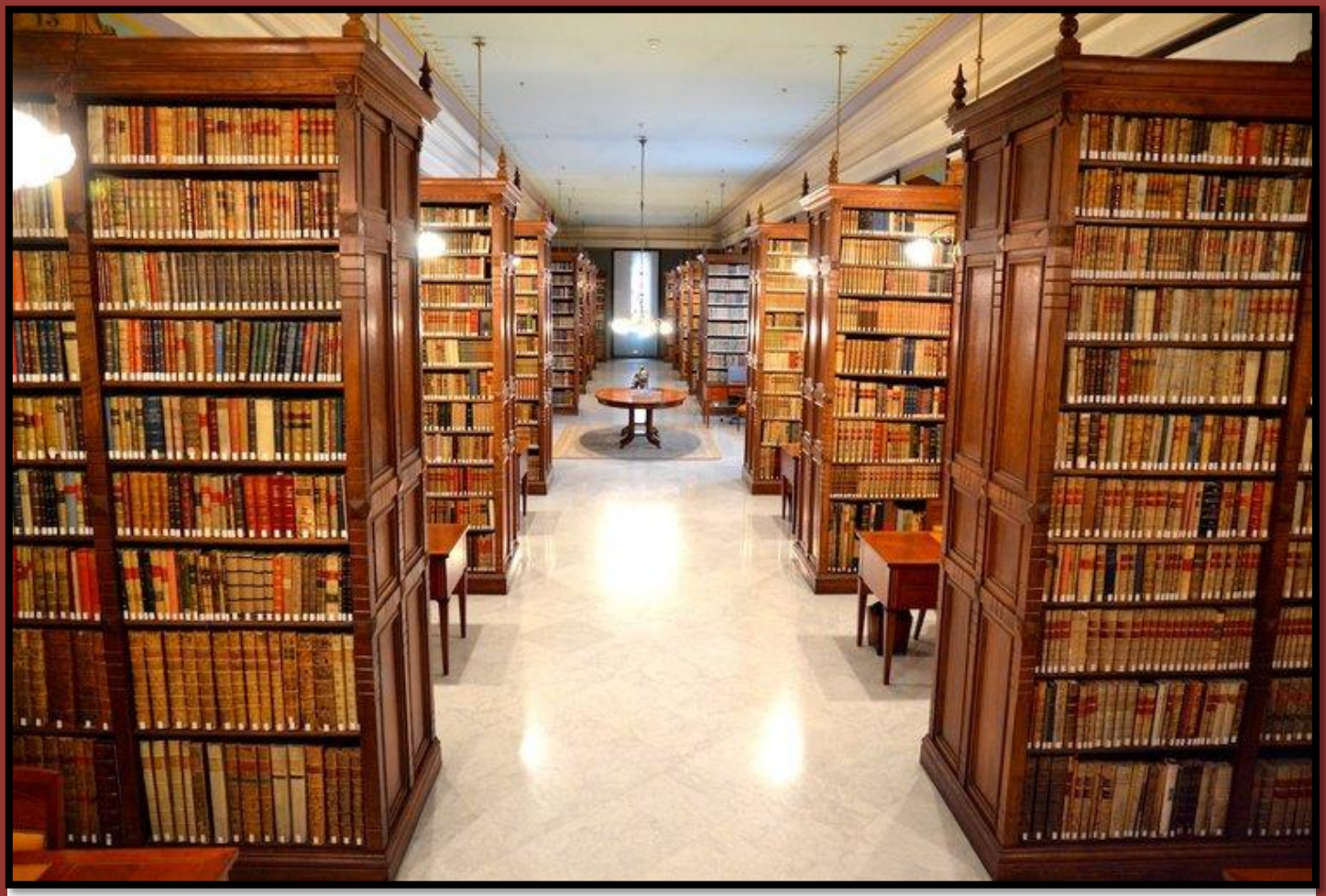


JOSE VIERA
DAVID SERRANO

BASE DE DATOS.

ÍNDICE.

1. Nombre de los integrantes del equipo.

Somos José Viera y David Serrano.

2. Descripción general de la solución aportada: Detallar en esta sección todas aquellas cuestiones importantes asumidas a la hora de resolver el proyecto.

Una biblioteca desea organizar su actividad diaria de forma sencilla, organizando los usuarios, empleados, libros y préstamos de la misma.

De manera que, de los usuarios sepan sus datos personales y ver en todo momento que libros han sido prestados, cuales les faltan por prestar, cuantos tiene ahora mismo.

De los empleados se desea conocer sus datos personales y conocer que prestamos realizo a que cliente.

De los libros se desea llevar de forma ordenada sus títulos, a que género pertenecen, quien es el/los autor/es, número de páginas que tienen, etc.

Por lo tanto, de los datos recibidos se asume que:

- El usuario, si está bloqueado, no puede volver a hacer un préstamo, hasta que haya sido desbloqueado.

- Un usuario estará desbloqueado cuando su valor sea 0, distinto de 0 será que está bloqueado.

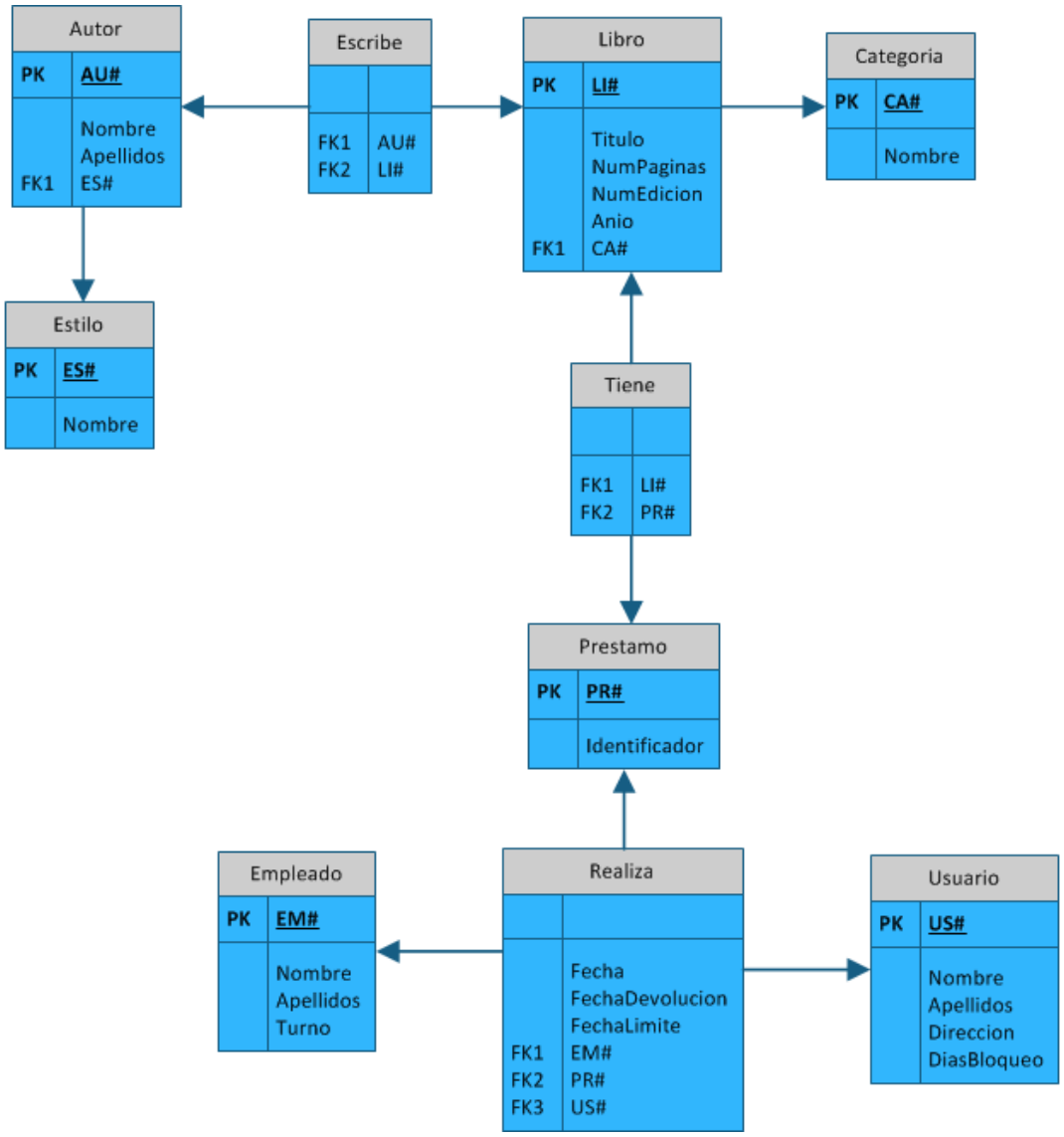
- Cada préstamo tendrá un identificador, con información precisa para distinguir entre los distintos préstamos.

- Un usuario tendrá 14 días para entregar un libro, pudiendo hacerlo antes de dicho plazo.

- De los autores, se deseará conocer su nombre y apellidos. Así como, de su estilo el nombre.

3. Diagrama de datos de la solución: Diagrama entidad – relación y tablas generadas a partir del diagrama entidad – relación.

Después de atender y estudiar las necesidades de la biblioteca, se organizó el siguiente diagrama.



4. Implementación sobre ORACLE del modelo planteado.

4.1. Definición de las tablas a crear (CREATE....).

Ya con el diagrama creado, se dispuso a crear las tablas. Hay que tener en cuenta que el orden es muy importante a la hora de implantar en una base de datos. Primero se crearán las tablas que no dependan de ninguna otra tabla (no tengan claves foráneas o referencias a otras tablas).

```
CREATE TABLE Estilo
(
        ES#    VARCHAR2(4),
        Nombre VARCHAR2(30),
        CONSTRAINT PK_Estilo_ES# PRIMARY KEY (ES#)
);
```

```
CREATE TABLE Autor
(
        AU#      VARCHAR2(4),
        Nombre    VARCHAR2(30),
        Apellidos VARCHAR2(50),
        ES# VARCHAR2(4),
        CONSTRAINT PK_Autor_AU# PRIMARY KEY (AU#),
        CONSTRAINT FK_Autor_ES# FOREIGN KEY (ES#) REFERENCES Estilo(ES#)
);
```

```
CREATE TABLE Categoria
(
        CA#    VARCHAR2(4),
        Nombre VARCHAR2(30),
        CONSTRAINT PK_Categoria_CA# PRIMARY KEY (CA#)
);
```

```
CREATE TABLE Libro
(
        LI#       VARCHAR2(4),
        Titulo     VARCHAR2(50),
        NumPaginas VARCHAR2(4),
        NumEdicion VARCHAR2(30),
        Anio      VARCHAR2(4),
        CA#       VARCHAR2(4),
        CONSTRAINT PK_Libro_LI# PRIMARY KEY (LI#),
        CONSTRAINT FK_Libro_CA# FOREIGN KEY (CA#) REFERENCES Categoria(CA#)
);
```

```
CREATE TABLE Escribe
(
        LI# VARCHAR2(4),
        AU# VARCHAR2(4),
        CONSTRAINT FK_Escribe_AU# FOREIGN KEY(AU#) REFERENCES Autor(AU#),
        CONSTRAINT FK_Escribe_LI# FOREIGN KEY(LI#) REFERENCES Libro(LI#)
);
```

```
CREATE TABLE Prestamo
(
        PR#         VARCHAR2(4),
        Identificador VARCHAR2(30),
        CONSTRAINT PK_Prestamo_PR# PRIMARY KEY (PR#)
);
```

```
CREATE TABLE Tiene
(
        LI# VARCHAR2(4),
        PR# VARCHAR2(4),
        CONSTRAINT FK_Tiene_LI# FOREIGN KEY(LI#) REFERENCES Libro(LI#),
        CONSTRAINT FK_Tiene_PR# FOREIGN KEY(PR#) REFERENCES Prestamo(PR#)
);
```

```
CREATE TABLE Empleado
(
        EM#      VARCHAR2(4),
        Nombre    VARCHAR2(30),
        Apellidos VARCHAR2(50),
        Turno     VARCHAR2(30),
        CONSTRAINT PK_Empleado_EM# PRIMARY KEY(EM#)
);
```

```
CREATE TABLE Usuario
(
        US#        VARCHAR2(4),
        Nombre      VARCHAR2(30),
        Apellidos   VARCHAR2(50),
        Direccion   VARCHAR2(30),
        DiasBloqueo NUMBER(4) DEFAULT 0,
        CONSTRAINT PK_Usuario_US# PRIMARY KEY(US#)
);
```

```
CREATE TABLE Realiza
(
        Fecha DATE,
        FechaDevolucion DATE,
        FechaLimite DATE,
        EM# VARCHAR2(4),
        PR# VARCHAR2(4),
        US# VARCHAR2(4),
        CONSTRAINT FK_Realiza_EM# FOREIGN KEY(EM#) REFERENCES Empleado(EM#),
        CONSTRAINT FK_Realiza_PR# FOREIGN KEY(PR#) REFERENCES Prestamo(PR#),
        CONSTRAINT FK_Realiza_US# FOREIGN KEY(US#) REFERENCES Usuario(US#)
);
```

Todos ellos, serán introducidos dentro de un script, para un manejo más sencillo a la hora de introducir las tablas en la base de datos.

También, por si ocurre algún problema a la hora de crear una tabla, se decide crear otro script para borrar las tablas, es el siguiente:

```
DROP TABLE Realiza;
DROP TABLE Empleado;
DROP TABLE Usuario;
DROP TABLE Tiene;
DROP TABLE Prestamo;
DROP TABLE Escribe;
DROP TABLE Autor;
DROP TABLE Estilo;
DROP TABLE Libro;
DROP TABLE Categoria;
```

Después

4.2. Expresión SQL empleada para resolver cada apartado (c),d)....l)).

c) Libros de la biblioteca organizados por categorías, conociendo para cada categoría el número total de libros.
Poder elegir la categoría y ver el detalle de los libros correspondientes.

```
SELECT Categoria.Nombre, COUNT(Libro.Titulo) AS "NUMERO DE LIBROS"
FROM Libro, Categoria
WHERE Libro.CA# = Categoria.CA#
GROUP BY Categoria.Nombre
ORDER BY Categoria.Nombre;

SELECT Li.Titulo, Li.NumPaginas, Li.NumEdicion, Anio, Ca.Nombre
FROM Libro Li JOIN Categoria Ca ON Li.CA# = Ca.CA#
WHERE Ca.Nombre = 'Novela';
```

NOTA: en la última sentencia, aunque pone "Novela" se podría introducir cualquier otra categoría que se elija.

e) Categoría con mayor cantidad de libros: Poder conocer el nombre de la categoría que mayor cantidad de libros contiene.

```
SELECT Categoria.Nombre, COUNT(Categoria.Nombre) AS "NUMERO DE LIBROS"
FROM Categoria, Libro
WHERE Categoria.CA# = Libro.CA#
GROUP BY Categoria.Nombre
HAVING COUNT(Categoria.Nombre) =
(SELECT MAX(contador) FROM
        (SELECT COUNT(Categoria.Nombre) contador
        FROM Categoria, Libro
```

```
        WHERE Categoria.CA# = Libro.CA#
        GROUP BY Categoria.Nombre));
```

f) Libros de la biblioteca organizados por estilos de el/los autor/es, conociendo para cada estilo el número total de libros.
Poder elegir el estilo y ver el detalle de los libros correspondientes.

```
SELECT Libro.Titulo, Estilo.Nombre AS "ESTILO"
FROM Libro, Autor, Escribe, Estilo
WHERE Estilo.ES# = Autor.ES# AND
        Autor.AU# = Escribe.AU# AND
        Escribe.LI# = Libro.LI#
ORDER BY Estilo.Nombre;

SELECT COUNT(Libro.Titulo) AS "NUMERO DE LIBROS", Estilo.Nombre AS "NOMBRE ESTILO"
FROM Libro, Autor, Escribe, Estilo
WHERE Estilo.ES# = Autor.ES# AND
        Autor.AU# = Escribe.AU# AND
        Escribe.LI# = Libro.LI#
GROUP BY Estilo.Nombre
ORDER BY Estilo.Nombre;

SELECT Libro.Titulo, Libro.NumPaginas, Libro.NumEdicion, Libro.Anio, Estilo.Nombre
FROM Libro, Autor, Escribe, Estilo
WHERE Estilo.ES# = Autor.ES# AND
        Autor.AU# = Escribe.AU# AND
        Escribe.LI# = Libro.LI# AND
        Estilo.Nombre = 'Periodismo';
```

NOTA: en la última sentencia, aunque pone "Periodismo" se podría introducir cualquier otra categoría que se elija.

g) Libros que se encuentran actualmente prestados (prestados, no devueltos y dentro del plazo de 2 semanas): Libro, Usuario, Fecha de préstamo.

```
SELECT Libro.Titulo, Usuario.Nombre, Usuario.Apellidos, Realiza.Fecha
FROM Libro, Tiene, Prestamo, Realiza, Usuario
WHERE Libro.LI#    = Tiene.LI#    AND
        Tiene.PR#    = Prestamo.PR# AND
        Prestamo.PR# = Realiza.PR#  AND
        Realiza.US#  = Usuario.US#  AND
        Realiza.FechaDevolucion IS NULL AND
        SYSDATE < FechaLimite;
```

h) Libros que se han prestado y se han devuelto (prestados y devueltos): Libro, Usuario, Fecha de préstamo, Fecha de devolución.

```
SELECT Libro.Titulo, Usuario.Nombre, Usuario.Apellidos, Realiza.Fecha,
Realiza.FechaDevolucion
FROM libro, Tiene, Prestamo, Realiza, Usuario
WHERE FechaDevolucion IS NOT NULL AND
        Libro.LI# = Tiene.LI# AND
        Tiene.PR# = Prestamo.PR# AND
        Prestamo.PR# = Realiza.PR# AND
        Realiza.US# = Usuario.US#;
```

i) Libros que están actualmente fuera de plazo (prestados, no devueltos y fuera del plazo de 2 semanas): Libro, Usuario, Fecha de préstamo, Fecha Actual.

```
SELECT Libro.Titulo, Realiza.Fecha, SYSDATE
FROM Libro, Tiene, Prestamo, Realiza
WHERE Libro.LI# = Tiene.LI# AND
        Tiene.PR# = Prestamo.PR# AND
        Prestamo.PR# = Realiza.PR# AND
        Realiza.FechaDevolucion IS NULL AND
        SYSDATE > Realiza.FechaLimite;
```

j) Préstamos efectuados por cada usuario, en el que aparezca para cada usuario, los préstamos que ha realizado indicando la fecha de retirada y devolución y que se detalle el/los libros que retiró.

```
SELECT Usuario.Nombre, Usuario.Apellidos, Prestamo.Identificador, Realiza.Fecha,
FechaDevolucion, Libro.Titulo
FROM Libro, Tiene, Prestamo, Realiza, Usuario
WHERE Libro.LI# = Tiene.LI# AND
        Tiene.PR# = Prestamo.PR# AND
        Prestamo.PR# = Realiza.PR# AND
        Realiza.US# = Usuario.US# AND
        FechaDevolucion IS NOT NULL
ORDER BY Usuario.Nombre, Realiza.Fecha;
```

k) Bloquear automáticamente a los usuarios que hayan excedido el período de devolución de libros.

```
--Usuarios con libros sin devolver y fuera de plazo.
UPDATE Usuario
SET  DiasBloqueo = (SELECT (TRUNC(SYSDATE - Realiza.FechaLimite) * 5 )
                                        FROM Realiza
                                        WHERE Realiza.US# = Usuario.US# AND
                                        FechaDevolucion IS NULL        AND
                                        SYSDATE > FechaLimite
                                        )
WHERE US# IN (SELECT US# FROM Realiza WHERE SYSDATE > FechaLimite AND
FechaDevolucion IS NULL);

--Usuarios con libros devueltos fuera de la fecha limite.
UPDATE Usuario
SET  DiasBloqueo = (SELECT (TRUNC(Realiza.FechaDevolucion - Realiza.FechaLimite) * 5 )
                                        FROM Realiza
                                        WHERE Realiza.US# = Usuario.US# AND
                                        FechaDevolucion > FechaLimite
                                        )
WHERE US# IN (SELECT US#
                        FROM Realiza
                        WHERE Realiza.US# = Usuario.US# AND
                                        FechaDevolucion > FechaLimite
                        );
```

5. Consideraciones finales y conclusiones.

Los resultados del presente trabajo se han estructurado en varias partes. La primera de presentarse con un modelo de las cosas que necesita un cliente (en este caso una biblioteca) y poder organizar un diagrama para su posible utilización. Segundo, creación de una base de datos con el modelo del diagrama delante. Después, el correcto manejo a través de la base de datos, y sacar la información que se necesita. Y, por último, la actualización de tablas ya creadas.