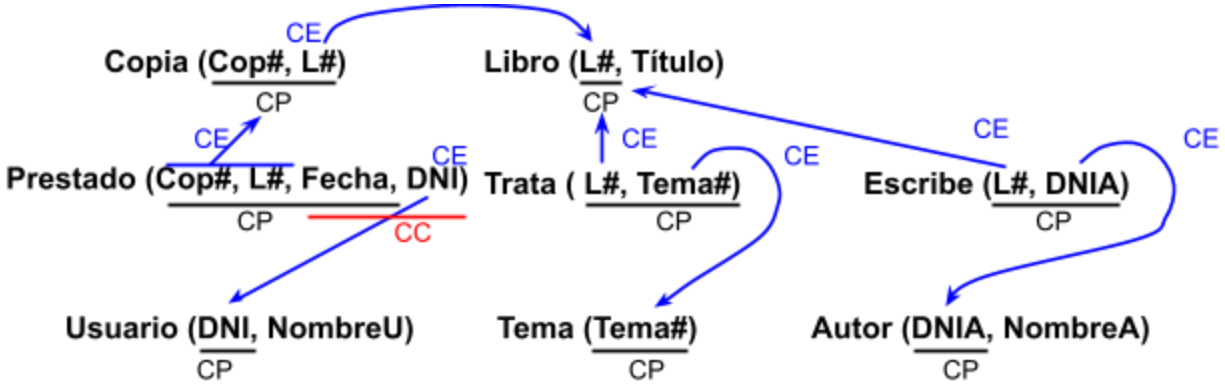
Fundamentos de Bases de Datos. Grupo A
Parcial 2. Ejercicio Práctico

Nombre Alumno:
Profesor del Grupo:

Disponemos de la siguiente BD que gestiona libros,copias y préstamos en una biblioteca:



Se pide:

A. Escribir las sentencias SQL para crear tablas `Copia` y `Prestado` suponiendo que el resto de las tablas ya están creadas. Debe tenerse en cuenta lo siguiente:
   a. Deben consignarse todas las restricciones indicadas en el esquema.
   b. Todos los campos son de tipo alfanumérico, salvo Fecha que es de tipo date.
   c. El año 2012 la biblioteca estuvo cerrada por reformas, por lo que debe evitarse que se introduzcan préstamos durante ese año. (**2.5 pts.**).

Res:
```
CREATE TABLE Copia(
Cop# VARCHAR2(10),
L# VARCHAR2(10) REFERENCES libro(L#),
PRIMARY KEY(Cop#,L#));

CREATE TABLE Prestado(
Cop# VARCHAR2(10),
L# VARCHAR2(10),
Fecha date CHECK (to_char(fecha,'YYYY')<>'2012'),
DNI VARCHAR2(9) REFERENCES usuario(DNI),
FOREIGN KEY (Cop#,L#) REFERENCES copia(Cop#,L#),
PRIMARY KEY(Cop#,L#,Fecha),
UNIQUE (Fecha,DNI));
```

B. Expresar mediante Álgebra y Cálculo las siguientes consultas (**1.25 pt.** cada una):
   a. "Encontrar el nombre de los usuarios que han tomado prestado todos los libros de la biblioteca".

$\pi_{nombreU}\ (usuario \bowtie (\pi_{\ DNI,L\#}(prestado) \div \pi_{\ L\#}(libro)))$

WinRDBI:

```
{U.nombreU | usuario(U) and not (exists L) (libro(L) and not (exists(P)
(prestado(P) and P.L#=L.L# and P.DNI=U.DNI))}
```

en **QUEL**:

```
RANGE U in usuario
RANGE P in prestado
RANGE L in libro
SELECT U.nombreU WHERE
∀L( ∃P(P.L#=L.L# ∧ P.DNI=U.DNI))
 ó
SELECT U.nombreU WHERE
∄L( ∄P(P.L#=L.L# ∧ P.DNI#=U.DNI))
```

b. "Mostrar el título del primer libro que se prestó (fecha de préstamo menor)".

$\rho(prestado) = p1$

$\rho(prestado) = p2$

$\pi_{Titulo}(libro \bowtie (\pi_{L\#}(prestado) - (\pi_{p1.L\#}(\sigma_{p1.fecha>p2.fecha}(p1 \times p2)))))$

```
{ L.titulo | libro(L) and (exists P1) (prestado(P1) and P1.L#=L.L# and
not (exists P2) (prestado(P2) and P1.fecha>P2.fecha)) }
```

en **QUEL**:

```
RANGE P1 ,P2 IN Prestado; RANGE L in libro
SELECT L.titulo WHERE
∃P1(∀P2(P1.L#=L.L# ∧ P1.Fecha<P2.Fecha))
ó
SELECT L.titulo WHERE
∃P1(∄P2(P1.L#=L.L# ∧ P1.Fecha>P2.Fecha))
```

C. Expresar mediante SQL la consulta a) del apartado anterior. (**1.25 pt**.).

```
SELECT u.nombreU FROM usuarios u WHERE NOT EXISTS (SELECT * FROM libro
l WHERE NOT EXISTS (SELECT * FROM prestado p WHERE p.L#=l.LV# AND
p.DNI=u.DNI));

SELECT u.nombreU FROM usuarios u WHERE NOT EXISTS (
(SELECT l.L# FROM libro l)
MINUS
(SELECT p.L# FROM prestado p WHERE p.DNI=u.DNI));
```

D. Crear la vista DNI_Libros_prestados que muestre la siguiente información: DNI de usuario, título de libros que ha tomado prestado y cuantas veces cada uno de ellos, para aquellos usuarios y títulos de libro que se hayan tomado prestado más de una vez, ordenados por DNI y título. (**1.25 pt**.).

```
CREATE VIEW DNI_Libros_prestados (DNI,Titulo,veces_prestado) AS (SELECT
p.DNI,l.titulo,count(*)  FROM prestado p,libro l WHERE p.L#=l.L# GROUP
BY (p.DNI,l.titulo) HAVING count(*) >1 ORDER BY p.DNI,l.titulo);
```

Tiempo total de realización: 1 hora y 30 minutos.