

## Solució Exercici Prac. 3b (Esquema).

---

LIBRO(codlib, título, autores, temática, totalpres)

CP= {codlib}

VNN= {título}

PRÉSTAMO(codlib, codsoc, fechapres, fechadev)

CP= {codlib, fechapres}

VNN= {codsoc}

CAj= {codlib} → LIBRO

CAj= {codsoc} → SOCIO

SOCIO(codsoc, nombre, dir, tel)

CP= {codsoc}

VNN= {nombre}

## Solució Exercici Prac. 3b (Esquema).

---

R1:

LX:LIBRO

$$\forall LX ((\text{LIBRO}(LX) \wedge (\neg \text{nulo}(LX.\text{temática}))) \\ \rightarrow ( (\text{LX}.\text{temática} = \text{'Física'}) \vee \\ (\text{LX}.\text{temática} = \text{'Óptica'}) \vee \\ (\text{LX}.\text{temática} = \text{'Mecánica'}) \vee \\ (\text{LX}.\text{temática} = \text{'Electricidad'}) ))$$

R2:

PX:PRÉSTAMO

$$\forall PX ((\text{PRÉSTAMO}(PX) \wedge (\neg \text{nulo}(PX.\text{fechadev}))) \\ \rightarrow (PX.\text{fechapres} \leq PX.\text{fechadev}))$$

Faltarien algunes restriccions, com que no hi haja dos préstecs del mateix llibre amb dates solapades.

# Solució Parcial Exercici (SQL).

---

CREATE TABLE socio

( codsoc NUMBER **constraint cp\_socio** PRIMARY KEY **deferrable**,  
nombre varchar(40) NOT NULL, dir varchar(60),  
tel varchar(15), totalpres number **default 0 not null deferrable** );

CREATE TABLE libro

(codlib number **constraint cp\_libro** PRIMARY KEY **deferrable**,  
titulo varchar(50) NOT NULL, autores varchar(60) NOT NULL,  
tematica varchar(20),  
**constraint r1** CHECK (tematica IN ('Fisica','Optica','Mecanica','Electricidad')) );

CREATE TABLE prestamo

(codlib number, codsoc number NOT NULL, fechapres date, fechadev date,  
**constraint cp\_prestamo** PRIMARY KEY(codlib, fechapres) **deferrable**,  
**constraint ca\_pre\_libro** FOREIGN KEY (codlib) REFERENCES libro **deferrable**,  
**constraint ca\_pre\_socio** FOREIGN KEY (codsoc) REFERENCES socio **deferrable**,  
**constraint R2** CHECK (fechapres <= fechadev) **deferrable** );

# Solució Exercici (SQL).

---

Queda mantindre la dada “totalpres”...

Solució ORACLE:  
TRIGGERS

# Solució Exercici (SQL).

ESDEVENIMENTS que han d'afectar a "totalpres":

ESDEVENIMENT			CONDICIÓ	ACCIÓ	
Operació	Taula	Atribut			
- E- P- R- E- S- E- V- E- N- I- M- E- N- T- S-	INSERT	PRESTAMO	-	nul·la(fechadev)	+1
	UPDATE	PRESTAMO	fechadev	de nul·la a no nul·la	-1
	UPDATE	PRESTAMO	codsoc	de no nul·la a nul·la	+1 (o no perm.)
	DELETE	PRESTAMO	-	nul·la(fechadev)	+1 i -1
INSERT	SOCIO	-	si $\leq$ 0	no permetre	
UPDATE	SOCIO	totalpres	si no correcte	no permetre	

## Solució Exercici (SQL).

---

Se ha d'afegir un valor per defecte a totalpres en la definició de la taula “socio” de la següent manera:

```
totalpres DEFAULT 0
```

I s'han de crear els següents triggers per a mantindre l'atribut derivat “totalpres”.

## Solució Exercici (SQL).

---

El següent disparador controla la inserció de nous préstecs:

```
CREATE TRIGGER T1
AFTER INSERT ON Prestamo
FOR EACH ROW
BEGIN
    IF :new.fechadev IS NULL THEN
        UPDATE socio SET totalpres = totalpres +1
            WHERE codsoc = :new.codsoc;
    END IF;
END;
```

## Solució Exercici (SQL).

---

El següent disparador controla la modificació de l'atribut “fechadev” de préstec (es torna o es modifica a prestat un llibre):

```
CREATE TRIGGER T2
AFTER UPDATE OF fechadev ON Prestamo
FOR EACH ROW
BEGIN
    IF :new.fechadev IS NULL AND :old.fechadev IS NOT NULL THEN
        UPDATE socio SET totalpres = totalpres +1 WHERE codsoc = :new.codsoc;
    ELSE IF :new.fechadev IS NOT NULL AND :old.fechadev IS NULL THEN
        UPDATE socio SET totalpres = totalpres -1 WHERE codsoc = :new.codsoc;
    END IF;
END IF;
END;
```



## Solució Exercici (SQL).

---

El següent disparador controla la modificació de l'atribut “codsoc” de préstecs (se li assigna un préstec a altres soci):

```
CREATE TRIGGER T3
AFTER UPDATE OF codsoc ON Prestamo
FOR EACH ROW
BEGIN
    IF :new.fechadev IS NULL THEN
        UPDATE socio SET totalpres = totalpres +1 WHERE codsoc = :new.codsoc;
        UPDATE socio SET totalpres = totalpres -1 WHERE codsoc = :old.codsoc;
    END IF;
END;
```

## Solució Exercici (SQL).

---

El següent disparador controla el esborrat de préstecs (només es pot fer si el llibre s'ha tornat):

```
CREATE TRIGGER T4
AFTER DELETE ON Prestamo
FOR EACH ROW
BEGIN
    IF :old.fechadev IS NULL THEN
        UPDATE socio SET totalpres = totalpres -1
            WHERE codsoc = :old.codsoc;
    END IF;
END;
```

## Solució Exercici (SQL).

---

El següent disparador controla que el número de préstecs sempre siga 0 en inserir un nou soci:

```
CREATE TRIGGER T5
AFTER INSERT ON Socio
FOR EACH ROW
BEGIN
    IF :new.totalpres <> 0 then
        RAISE_APPLICATION_ERROR(-20000, 'Quan s'insereix un préstec,
            el seu número de préstecs ha de ser 0.');
```

```
    END IF;
END;
```

# Solució Exercici (SQL).

---

El següent disparador controla que el número de préstecs d'un soci no es puga modificar:

```
CREATE TRIGGER T6
AFTER UPDATE OF totalpres ON Socio
FOR EACH ROW
DECLARE
  N NUMBER;
BEGIN
  SELECT COUNT(*) INTO N
  FROM PRESTAMO P
  WHERE P.fechadev IS NULL AND P.codsoc = :new.codsoc;
  IF N <> :new.totalpres THEN
    RAISE_APPLICATION_ERROR(-20000, 'Modificació no consistent.');
```

Aquest trigger podria donar problemes perquè pot ser cridat per altres triggers (T1-T4).

Una opció millor és no permetre la modificació de *totalpres* a cap usuari utilitzant permisos:

```
REVOKE UPATE(totalpres) TO ALL;
```