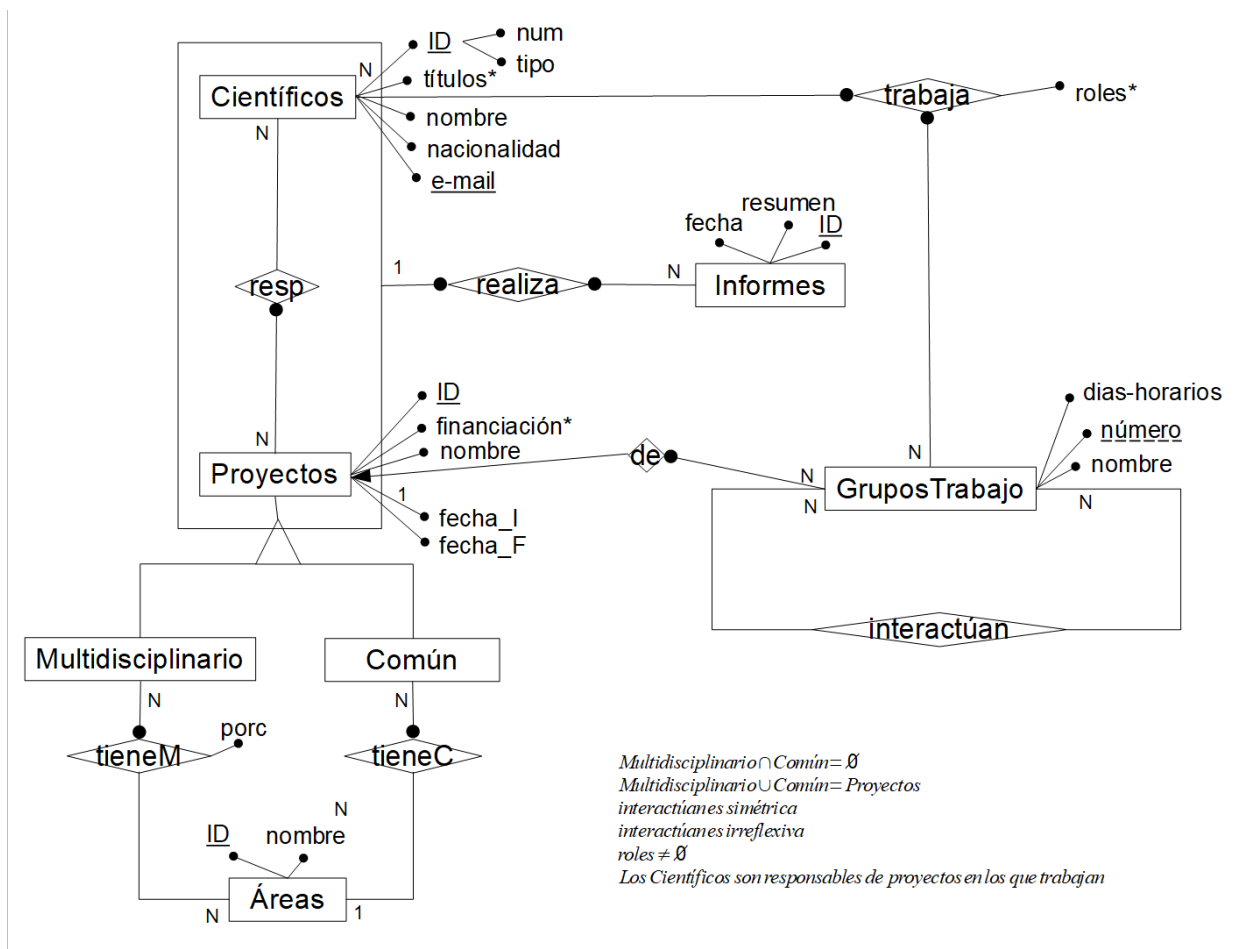


## FUNDAMENTOS DE BASES DE DATOS

### Examen Julio 2021

#### Ejercicio 1 (25 puntos).



## Ejercicio 2 (25 puntos).

Considere una biblioteca donde se desea registrar la información de los socios, de los empleados, de los libros existentes en la biblioteca, de los préstamos realizados y de las compras de libros que realiza la biblioteca.

De cada socio se conoce su cédula, teléfono, dirección, y un conjunto de emails. De cada empleado se conoce su cédula, nombre y teléfono. De cada libro se guarda un código (que lo identifica), un título y la cantidad de copias. De cada préstamo que se realiza a un socio se registra el socio, el libro, la fecha y el plazo del préstamo. De cada compra realizada se registra el libro, el empleado que realizó la compra, la fecha de la compra y la cantidad comprada.

Las cédulas se utilizan como identificadores de las personas.

Se asume que en una misma fecha, no se realizarán dos compras de un mismo libro. Además, se asume que en una misma fecha el socio no saca en préstamo más de una vez el mismo libro, por lo tanto, a una fecha de préstamo, un libro y un socio les corresponde un sólo plazo.

Se utilizan los siguientes nombres de atributos:

ci-socio	nom-emp	fecha-pres
tel-socio	tel-emp	plazo-pres
direccion	cod-libro	fecha-comp
email	tit-libro	cant-comp
ci-emp	cant-copias	

### Se pide:

- Deducir las dependencias funcionales que deberían cumplirse para esta realidad.
- Para la Relación Universal RU y el conjunto F de dependencias funcionales hallado en la parte a), hallar un cubrimiento minimal de F y las claves de RU, justificando.
- Partiendo de RU, dar una descomposición en 3NF, aplicando el algoritmo visto en clase.
- Decir si la descomposición obtenida preserva dependencias funcionales y tiene JSP.
- Hallar una dependencia multivaluada embebida que se cumpla en alguna de las relaciones obtenidas en la parte c). Justificar.
- Ahora, suponiendo que se llegó a la siguiente descomposición de la relación universal:

**Socios (ci-socio, tel-socio, direccion, email)**  
**Empleados (ci-emp, nom-emp, tel-emp)**  
**Libros (cod-libro, tit-libro, cant-copias)**  
**Préstamos (ci-socio, cod-libro, fecha-pres, plazo-pres)**  
**Compras (ci-emp, cod-libro, fecha-comp, cant-comp)**

decir en qué forma normal se encuentra la descomposición, justificando.

**SOLUCION:**

**a)**

ci-socio → tel-socio, direccion

ci-emp → nom-emp, tel-emp

cod-libro → tit-libro, cant-copias

cod-libro, fecha-comp → ci-emp, cant-comp

ci-socio, fecha-pres, cod-libro → plazo-pres

**b)**

**RelUniversal** (ci-socio, tel-socio, direccion, email, ci-emp, nom-emp, tel-emp, cod-libro, tit-libro, cant-copias, fecha-pres, fecha-comp, cant-comp, plazo-pres)

Conjunto de dfs minimal:

$F = \{$   
ci-socio → tel-socio  
ci-socio → direccion  
ci-emp → nom-emp  
ci-emp → tel-emp  
cod-libro → tit-libro  
cod-libro → cant-copias  
cod-libro, fecha-comp → ci-emp  
cod-libro, fecha-comp → cant-comp  
ci-socio, fecha-pres, cod-libro → plazo-pres  
 $\}$

F es minimal porque cumple las 3 propiedades:

(1) Todas las dfs tienen un solo atributo a la derecha.

(2) Ninguna df tiene atributos redundantes a la izquierda. Esto lo verificamos en las últimas 3 dfs. En cada una de ellas se cumple que si sacamos cualquier atributo de la izquierda, no se puede obtener el de la derecha.

(3) No hay dfs redundantes. Esto se verifica muy fácilmente, viendo que todos los atributos de la derecha de las dfs aparecen una sola vez a la derecha, por lo tanto si elimino una df no voy a poder llegar a determinar el atributo de la derecha de esa df.

Unica clave: ci-socio, cod-libro, fecha-pres, fecha-comp, email

Es clave porque a partir de ese conjunto de atributos determino a todos los demás atributos de RelUniversal y si quito a cualquiera de ellos, ya no determino a todos. Es la única clave porque cada uno de esos atributos tienen que estar en todas las claves, ya que no aparecen a la derecha de ninguna df.

**c)**

Aplicamos siguientes pasos del algoritmo de 3NF con JSP y preservación de dfs:

Construir relaciones a partir de las dfs:

**Socios (ci-socio, tel-socio, direccion)**  
**Empleados (ci-emp, nom-emp, tel-emp)**  
**Libros (cod-libro, tit-libro, cant-copias)**  
**Prestamos (ci-socio, fecha-pres, cod-libro, plazo-pres)**  
**Compras (cod-libro, fecha-comp, ci-emp, cant-comp)**

Agregar una relación con una clave:

**Pres-Comp (ci-socio, cod-libro, fecha-pres, fecha-comp, email)**

d)

La descomposición preserva las dependencias funcionales y tiene JSP, porque esto lo asegura el algoritmo aplicado.

e)

Algunas dependencias multivaluadas embebidas que se cumplen en la relación Pres-Comp son las siguientes:

ci-socio -->> cod-libro | email

cod-libro -->> fecha-pres | fecha-comp

f)

Descomposición:

**Socios (ci-socio, tel-socio, direccion, email)**  
**Empleados (ci-emp, nom-emp, tel-emp)**  
**Libros (cod-libro, tit-libro, cant-copias)**  
**Prestamos (ci-socio, cod-libro, fecha-pres, plazo-pres)**  
**Compras (ci-emp, cod-libro, fecha-comp, cant-comp)**

La forma normal de la descomposición planteada es la forma normal menos restrictiva de las de los sub-esquemas de la descomposición.

Forma normal de **Socios (ci-socio, tel-socio, direccion, email)**:

Proyectamos las dfs halladas en la parte (a) sobre la relación Socios.

$F_{\text{Socios}} = \{ \text{ci-socio} \rightarrow \text{tel-socio, direccion} \}$

Clave: (ci-socio, email)

La df ci-socio  $\rightarrow$  tel-socio, direccion viola 2NF, ya que es una dependencia parcial de una clave.

Por lo tanto, la relación Socios está en 1NF.

No es necesario hallar las formas normales de los otros sub-esquemas de la descomposición, ya que 1NF es la forma normal menos restrictiva posible. **Por lo tanto, la descomposición se encuentra en 1NF.**

### Ejercicio 3 (25 puntos)

Un sistema de gestión de un gimnasio maneja una base de datos con el siguiente esquema:

#### INSTRUCTORES (ci, nombre, esDueño)

Contiene información básica sobre los instructores que trabajan en el gimnasio. El atributo *esDueño* indica si el instructor es uno de los dueños del gimnasio o no.

#### MIEMBROS (ci, nombre, edad)

Contiene información sobre los miembros que están inscriptos en el gimnasio.

#### CLASES (tipoClase, fecha, hora, cantAsistentes, cilnstructor)

Contiene información sobre las clases dictadas. El atributo *cilnstructor* es la cédula del instructor que dictó la clase, y el atributo *tipoClase* podría ser por ejemplo “crossfit” o “levantamiento olímpico”.

#### RESERVAS (tipoClase, fecha, hora, ciMiembro)

Contiene las reservas que realizan los miembros para las clases.

$$\begin{aligned} \Pi_{\text{cilnstructor}}(\text{CLASES}) &\subseteq \Pi_{\text{ci}}(\text{INSTRUCTORES}) \\ \Pi_{\text{ciMiembro}}(\text{RESERVAS}) &\subseteq \Pi_{\text{ci}}(\text{MIEMBROS}) \\ \Pi_{\text{tipoClase, fecha, hora}}(\text{RESERVAS}) &\subseteq \Pi_{\text{tipoClase, fecha, hora}}(\text{CLASES}) \end{aligned}$$

a) Resolver en **Cálculo Relacional**:

Devolver la cédula de los miembros que hayan reservado todas las clases dictadas por el instructor “Lee” a las cuales asistieron exactamente 10 personas.

$$\begin{aligned} \{ \langle m.ci \rangle / \text{MIEMBROS}(m) \wedge (\forall c) (\text{CLASES}(c) \wedge c.cantAsistentes = 10 \wedge \\ (\exists i)(\text{INSTRUCTORES}(i) \wedge i.ci = c.cilnstructor \wedge i.nombre = \text{“Lee”}) \rightarrow \\ (\exists r)(\text{RESERVAS}(r) \wedge r.ciMiembro = m.ci \wedge c.tipoClase = r.tipoClase \wedge \\ c.fecha = r.fecha \wedge c.hora = r.hora)) \wedge \\ (\exists u) (\text{CLASES}(u) \wedge u.cantAsistentes = 10 \wedge \\ (\exists j)(\text{INSTRUCTORES}(j) \wedge j.ci = u.cilnstructor \wedge j.nombre = \text{“Lee”})) \} \end{aligned}$$

b) Resolver en **Algebra Relacional**:

Devolver todos los datos de las clases de tipo “crossfit” que no fueron dictadas por el instructor “Franco”.

$$\begin{aligned} A &= \Pi_{\text{tipoClase, fecha, hora, cantAsistentes, cilnstructor}}(\text{CLASES} \mid_{\text{cilnstructor=ci}} (\sigma_{\text{nombre=“franco”}}(\text{INSTRUCTORES}))) \\ \text{Sol} &= \sigma_{\text{tipo=“crossfit”}}(\text{CLASES}) - A \end{aligned}$$

c) Resolver en **SQL**:

Devolver cédula y nombre de los miembros que hayan reservado dos clases en un mismo día al menos una vez.

```
SELECT DISTINCT m.ci, m.nombre
FROM MIEMBROS m JOIN RESERVAS r ON ci=ciMiembro
GROUP BY m.ci, m.nombre, r.fecha
HAVING count(*) > 1
```

d) ¿Se puede resolver la consulta de la parte c) en **Cálculo Relacional**? De ser posible resuélvala, de lo contrario justifique por qué no se puede.

Sí se puede.

$\{ \langle m.ci, m.nombre \rangle / MIEMBROS(m) \wedge (\exists r1) (RESERVAS(r1) \wedge r1.ciMiembro = m.ci \wedge (\exists r2) (RESERVAS(r2) \wedge r2.ci = m.ci \wedge r1.fecha = r2.fecha \wedge \neg(r1.tipoClase = r2.tipoClase \wedge r1.hora = r2.hora))) \}$

## Ejercicio 4 (25 puntos)

Dadas las transacciones:

**T1:** r1(x) w1(x) r1(y) r1(z) c1  
**T2:** r2(x) w2(x) w2(z) r2(z) c2

### Parte a)

Para las siguientes historias, indicar si son serializables, recuperables, evitan abortos en cascada y estrictas. **Justifique sus respuestas.**

**H1:** r1(x) w1(x) r2(x) r1(y) w2(x) r1(z) w2(z) r2(z) c2 c1

**H2:** r2(x) r1(x) w2(x) w1(x) r1(y) r1(z) c1 w2(z) r2(z) c2

**H3:** r2(x) r1(x) w2(x) w2(z) r2(z) c2 w1(x) r1(y) r1(z) c1

### Parte b)

Escriba T1, incluyendo las operaciones de read y lock necesarias, de forma de que cumpla con los requisitos planteados. **Justifique sus respuestas.**

- Asegurar de que siga 2PL estricto, pero no riguroso.
- Asegurar de que siga 2PL conservador
- Aplique alguna versión de 2PL que asegure de que no se generen deadlocks.

### Parte c)

Siendo **H** cualquier historia formada por las transacciones **T1** y **T2**, indicar si las siguientes afirmaciones son verdaderas (**V**) o falsas (**F**). **Justifique sus respuestas.** En caso de que sea falsa, use un contraejemplo a modo de justificación.

- Si T1 y T2 siguen 2PL conservador, se puede garantizar de que la historia es serializable y recuperable
- Si T1 y T2 siguen 2PL estricto entonces toda H es recuperable
- Si H es estricta, se puede asegurar que además evita abortos en cascada