

PREGUNTAS CORTAS

[Se valorará especialmente la capacidad de síntesis, con respuestas claras, breves y bien estructuradas]

Pregunta 1ª (1,5 puntos)

- a) Decir cuando un esquema de relación $R(A, DF)$ no está en cada una de las siguientes formas normales: 2FN, 3FN y FNBC.
- b) Mostrar un ejemplo de un esquema de relación que esté en 2FN pero que no esté en 3FN.

SOLUCION

- a) Un esquema de relación $R(A, DF)$ no está en :
 - 2FN. Cuando existen atributos no principales dependientes de parte de la clave.
 - 3FN. Cuando existe un atributo no principal dependiendo de otro no principal.
 - FNBC. Cuando existe alguna dependencia funcional cuyo determinante no es una clave.
- b)
 $R(AT, DEP)$ donde:
 $AT = \{A, B, C\}$ $DEP = \{B \rightarrow C, A \rightarrow B\}$ y $PK = (A)$

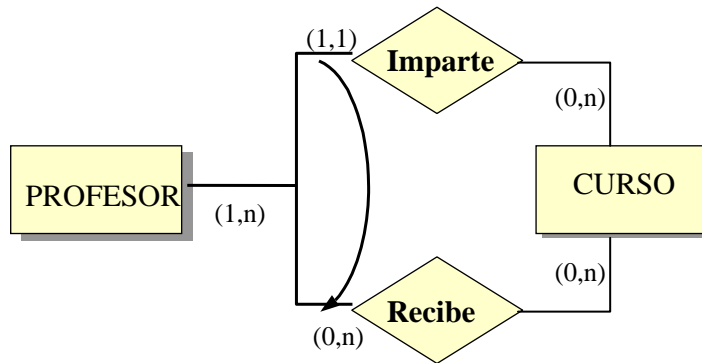
(utilizar únicamente el espacio anterior con letra clara y legible)

Pregunta 2ª (1,5 punto)

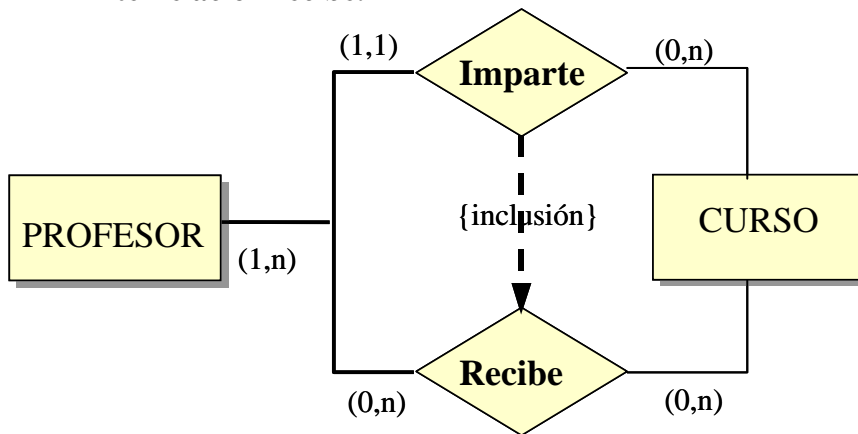
Realizar un ejemplo de las restricciones de inclusión y de inclusividad en el modelo E/R, dibujando el diagrama E/R correspondiente y explicando en lenguaje natural la restricción del mundo real que se quiere modelar a través de cada ejemplo.

SOLUCIÓN:

- **Restricción de Inclusividad:** si un profesor participa en **imparte** tiene necesariamente que participar en **recibe**

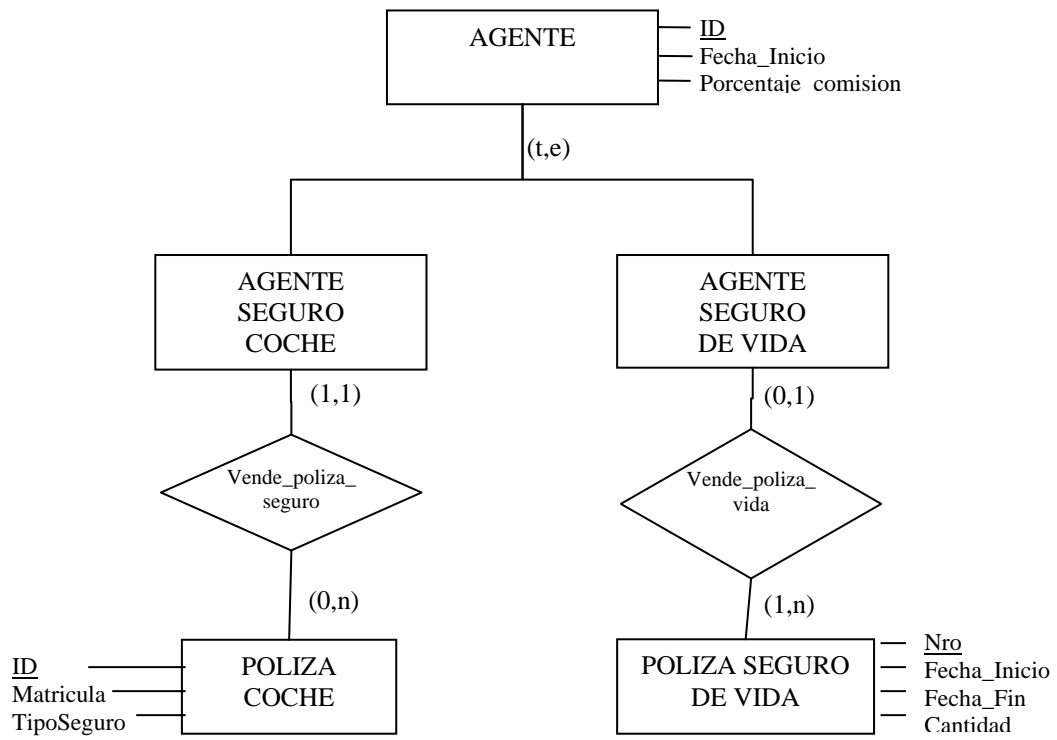


- **Restricción de Inclusión:** todo ejemplar de profesor que esté unido a un ejemplar de curso mediante la interrelación **imparte** tiene necesariamente que estar unido al mismo ejemplar de curso mediante la interrelación **recibe**.



Problema 1 (3,5 puntos)

1, Transformar a relacional el siguiente esquema E/R (1 punto):



SOLUCIÓN:

Agente(ID, Fecha_Inicio, Porcentaje_Comision)

AgenteSeguroCoche (ID)

AgenteSeguroVida (ID)

PolizaCoche (ID, Matricula, TipoSeguro, IDAgente)

PolizaSeguro (Nro, Fecha_Inicio, Fecha_Fin, Cantidad, IdAgente)

AgenteSeguroCoche.ID → Agente

AgenteSeguroVida → Agente

PolizaCoche.IDAgente → AgenteSeguroCoche

PolizaSeguro.IDAgente → AgenteSeguroVida

Restricción total y exclusividad:

$$\forall a, \text{Agente}(a) \rightarrow \exists asc, \neg \exists asv, \text{AgenteSeguroCoche}(asc) \wedge \text{AgenteSeguroVida}(asv) \wedge$$

$$a.ID = asc.ID \wedge$$

$$a.ID <> asv.ID$$

$$\vee$$

$$\exists asv, \neg \exists asc, \text{AgenteSeguroCoche}(asc) \wedge \text{AgenteSeguroVida}(asv) \wedge$$

$$a.ID <> asc.ID \wedge$$

$$a.ID = asv.ID$$

Todo agente de seguro de vida tiene, al menos, una póliza de seguro:

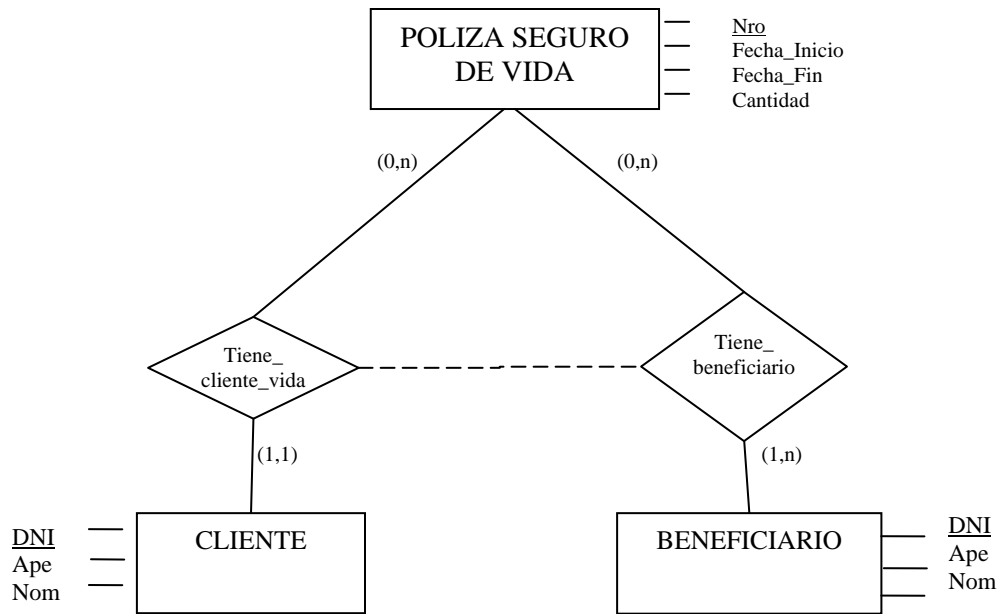
$$\forall asv, \text{AgenteSeguroVida}(asv) \rightarrow \exists ps, \text{PolizaSeguro}(ps) \wedge$$

$$asv.ID = ps.IDAgente$$

Toda póliza de coche la vende uno y sólo un agente de seguro de coche

PolizaCoche.IDAgente NOT NULL

2. Indicar si la transformación a relacional del siguiente modelo E/R tiene algún error y, en caso de tenerlo indicar cual y porqué (1 punto):



a) Transformación de entidades e interrelaciones

a.1) PolizaSeguroVida (Nro, Fecha_Inicio, Fecha_Fin, Cantidad, DNI_Cliente)
 DNI_Cliente → Cliente

a.2) TieneBeneficiario(Nro, Fecha_Inicio, DNI)
 Nro, Fecha_Inicio → PolizaSeguroVida
 DNI → Beneficiario

a.3) Beneficiario (DNI, Ape, Nom)

b) Restricciones de cardinalidad

b.1) Una póliza de seguro de vida tiene un y sólo un cliente
 DNI_Cliente NOT NULL

b.2) Una póliza de seguro de vida tiene, al menos, un beneficiario
 $\forall t, \text{TieneBeneficiario}(t) \rightarrow \exists b, \text{Beneficiario}(b) \wedge t.\text{DNI}=b.\text{DNI}$

c) Restricción de exclusión

$\forall p, \text{PolizaSeguroVida}(p) \rightarrow$
 $\exists tb, \text{TieneBeneficiario}(tb) \wedge p.\text{DNICliente} = \text{NULL}$
 OR
 $p.\text{DNICliente} \diamond \text{NULL} \wedge \neg \exists tb, \text{TieneBeneficiario}(tb)$

SOLUCIÓN:

- a.1) Error en la clave primaria de PolizaSeguroVida. Debe de ser Nro.
 - a.2) Error, en la clave primaria de TieneBeneficiario. Ya que este debe de estar compuesta por la clave de PolizaSeguroVida (que es Nro) y por la clave de Beneficiario (que es DNI).
 - a.3) Correcta
 - b.1) Correcta
 - b.2) Correcta
- c) Error. Está modelada una exclusividad y no una exclusión ya que lo que modela es que si póliza está relacionada con cliente entonces no puede estarlo con beneficiario

$\forall p, \text{PolizaSeguroVida}(p) \rightarrow$
 $\exists tb, \text{TieneBeneficiario}(tb) \wedge p.\text{DNICliente} = \text{NULL} \text{ /* lo está con beneficiario}$
 OR
 $p.\text{DNICliente} \neq \text{NULL} \wedge \neg \exists tb, \text{TieneBeneficiario}(tb) \text{ /*lo está con cliente}$

3. Sobre el esquema relacional siguiente (1, 5 puntos):

ComunidadVecinos(cod, nombre, calle, codpostal, población)
 Gestiona(numcolegiado, codcomunidad, honorarios)
 Administrador(numcolegiado, DNI, nombre)
 Compañía(CIF, nombre, direccion, sector, contacto, teléfono)
 Contrata(codcomunidad, CIF)

Gestiona.codcomunidad \rightarrow ComunidadVecinos
 Gestiona.numcolegiado \rightarrow Administrador
 Contrata.codcomunidad \rightarrow ComunidadVecinos
 Contrata.CIF \rightarrow Compañía

Realizar las siguientes consultas:

- a) En álgebra relacional, devolver el nombre de los administradores que administran comunidades sólo de de Ciudad Real.
- b) En cálculo relacional de dominio, devolver el nombre de las compañías que nunca fueron contratadas por ninguna comunidad.
- c) En cálculo relacional de tuplas, devolver el nombre de las compañías que son contratadas por comunidades que son gestionadas por administradores con honorarios superiores a 1200 euros.

SOLUCIÓN:

- a) En álgebra relacional, Devolver el nombre de los administradores que administran sólo comunidades de Ciudad Real

$\Pi_{\text{nombre}} (\text{Administrador} - (\text{Administrador} * \text{numcolegiado} (\Pi_{\text{poblacion} \neq \text{'Ciudad Real'}} (\text{ComunidadVecinos}) * \text{cod}=\text{codcomunidad} \text{ Gestiona})))$

A todos los administradores les quito aquellos que administran alguna comunidad distinta de Ciudad Real (aunque también puede administrar de Ciudad Real y de fuera de Ciudad Real) y me quedan los que administran sólo de Ciudad Real

- b) En cálculo de dominios. Devolver el nombre de las compañías que no han sido contratadas por ninguna comunidad

{ nombre | Compañía (CIF, nombre, -, -, -, -) \wedge \neg Contrata(-, CIF)}

- c) En cálculo de tuplas, Devolver el nombre de las compañías que son contratadas por comunidades que son gestionadas por administradores con honorarios superiores a 1200 euros

$\text{cia.nombre} \mid \text{Compañía}(\text{cia}) \wedge \exists \text{cv}, \exists \text{cia}, \exists \text{c}, \exists \text{g} (\text{ComunidadVecinos}(\text{cv}) \wedge \text{Compañía}(\text{cia}) \wedge \text{Contrata}(\text{c}) \wedge \text{Gestiona}(\text{g}) \wedge \text{cia.CIF} = \text{c.CIF} \wedge \text{c.codcomunidad} = \text{cv.cod} \wedge \text{cv.cod} = \text{g.codcomunidad} \wedge \text{g.honorarios} > 1200)$

Problema 2 (3,5 puntos)

Dado el siguiente esquema relacional de una biblioteca y sus préstamos:

Biblioteca(signatura, autor, titulo, editorial, clase_libro, carnet, tipo_carnet, nom_usuario, dir_usuario, tiempo_prestamo, fecha_inicio, fecha_fin, editorial, ciudad_edit, pais_edit)

Para simplificar realizamos las siguiente equivalencias:

Sig = signatuura; aut=autor; tit=titulo; edit=editorial; cla=clase_libro; car=carnet_usuario; tipo=tipo_carnet; nom=nom_usuario; dir=dir_usuario; tiempo=tiempo_prestamo; fi=fecha_inicio; ff=fecha_fin; edit=editorial; ciu=ciudad_edit; pais=pais_edit

Con las siguientes dependencias funcionales, que son un recubrimiento redundante:

```
DF: {
  f1. sig→aut
  f2. sig→tit
  f3. sig→edit
  f4. sig→cla
  f5. tit, aut→sig
  f6. car→tipo
  f7. car→nom
  f8. car→dir
  f9. cla→tiempo
  f10. sig, fi → car
  f11. car, fi→sig
  f12. car, fi→ff
  f13. edit→ciu
  f14. edit→pais
}
```

- 1) Contestar verdadero o falso, justificando la respuesta. (1 punto, 0,25 cada una)
 - a. La signatura de un libro determina el tiempo de préstamo del mismo.
Verdadero. Signatura→clase_libro y clase_libro→tiempo_prestamo entonces por transitividad signatura→tiempo_prestamo
 - b. Los atributos {titulo, autor} son equivalentes con el atributo signatura.
Verdadero. Titulo, autor→ signatura y signatura→autor, titulo
 - c. El carnet del usuario y la fecha de inicio determina el título y el autor
Verdadero. carnet_usuario, fecha_inicio→ signatura y signatura→ titulo, autor
 - d. Cada usuario puede tener más de un libro prestado en una determinada fecha.
Falso. Pues según la dependencia Carnet_usuario, fecha_inicio→ signatura, en una determinada fecha un usuario determinado sólo puede tener una signatura (un libro prestado)
- 2) Encontrar las claves (1 pto)
- 3) Encontrar una descomposición del esquema Biblioteca, utilizando el algoritmo de análisis, perdiendo el menor número de dependencias funcionales. (1,5 ptos)

SOLUCION

- 1) Encontrar las claves

Paso 1. Atributos independientes: no hay atributos independientes RSI=R

Paso 2. Busquemos atributos equivalentes

(tit,aut) ↔ sig

Nos quedamos con sig

Rsie=(sig,edit,cla,car,tipo,nom,dir,tiempo,fi,ff,ciu,pais)

DF = f3,f4,f6,f7,f8,f9,f10,f11,f12,f14 (quitamos f1, f2, f5)

- F3 sig→edit
- F4 sig→cla
- F6 car→tipo
- F7 car→nom
- F8 car→dir
- F9 cla→tiempo
- F10 sig, fi → car
- F11 car, fi→sig
- F12 car, fi→ff
- F13 edit→ciu
- F14 edit→pais

Paso 3.

Formamos Kp con los atributos que están sólo del lado izquierdo

Kp=sig,fi

Kp+=sig,fi,aut,tit,edit,cla,car,tipo,nom.dir,ff,tiempo,ciu,pais

Por lo tanto KP es la clave de Rsie

Paso 5. no hay independientes

Paso 6, reemplazo por los equivalentes

Las claves son K1= sig,fi, K2 = tit,aut,fi

3)

Agrupamos las dependencias con igual implicante

G1: {sig→aut,tit,edit,clase, tit,aut→sig}

G2 : {car→tipo,nom,dir}

G3: {clase→tiempo}

G4: {sig,fi→car}

G5: {car,fi→sig,ff}

G6: {edit→ciu,pais}

Cojo G2			
R1 {car,tipo,nom,dir} car→tipo,nom,dir K=car	R2 {sig,aut,tit,edit,cla,car,tiempo,fi,ff,ciu,pais} G1, G3 , G4, G5, G6		
	Cojo G3		
	R3 {cla, tiempo} cla→tiempo K = cla	R4 {sig,aut,tit,edit,cla,car,fi,ff,ciu,pais} G1, G4, G5, G6 NO ESTA FNBC	
		Cojo G6	
		R5 {edit,ciu,pais} edit→ciu,pais K=edit	R6 {sig,aut,tit,edit,cla, car,fi,ff} G1, G4, G5 NO ESTA EN FNBC
			Cojo G1
		R7 {sig,tit,aut,edit, cla} sig→aut,tit,edit, cla tit,aut→sig K1=sig, K2=tit, aut	R8 {sig,fi,ff,car} sig,fi→car car,fi→sig,ff K1={sig,fi} K2={car,fi} Está en FNBC

La solución es: R1, R3, R5, R7, R8