



**Examen de
Bases de datos y sistemas de información
I PARCIAL**

1) (0,7 puntos) Traducir a SQL la siguiente consulta QBE:

<i>A</i>	<i>C</i>	<i>S</i>	<i>I</i>
	<u><i>y</i></u>	s1	<u><i>z</i></u>

<i>B</i>	<i>N</i>	<i>C</i>
	<u><i>x</i></u>	<u><i>y</i></u>

<i>Resultado</i>	<i>N</i>	<i>C</i>	<i>I</i>
P.	AO(1). <u><i>x</i></u>	DO(2). <u><i>y</i></u>	<u><i>z</i></u>

Solución (10 min):

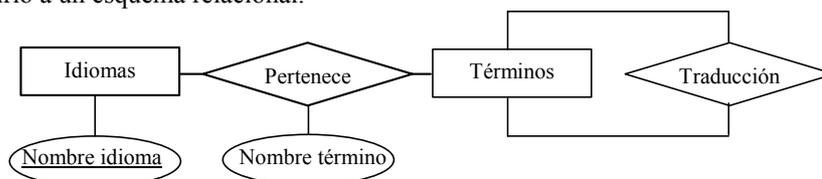
```
SELECT N, A.C, I
FROM A,B
WHERE A.C=B.C AND S='s1'
ORDER BY N ASC, A.C.DESC;
```

Otra posibilidad:

```
SELECT N,C,I
FROM A NATURAL INNER JOIN B
WHERE S='s1'
ORDER BY N ASC, A.C.DESC;
```

2) (1 punto) El siguiente esquema entidad-relación pretende representar términos (con un literal 'Nombre término'), cada uno de los cuales pertenece a un idioma (identificado por el literal 'Nombre idioma') y tiene una traducción (otro término) en otro idioma. Sin embargo, es incorrecto. Se pide:

- Indicar por qué.
- Reescribirlo en otro correcto sin añadir atributos nuevos.
- Traducirlo a un esquema relacional.



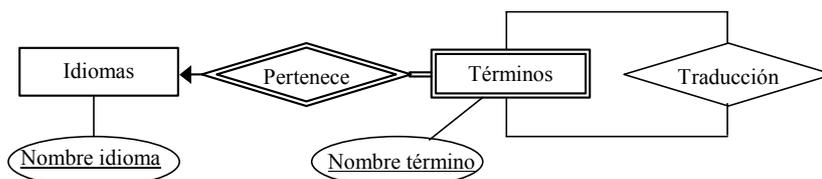
Solución (15 min):

a)

La entidad Términos no tiene ningún atributo que pueda identificarlo unívocamente y tampoco se trata de una entidad débil que pueda tomarlos prestados.

b)

Puesto que no se pueden añadir atributos, la única forma de poder identificar a Términos es convirtiéndola en una entidad débil. Sin embargo, Nombre idioma no es suficiente para identificar unívocamente a cada término (se necesitaría saber además su lexema (Nombre término)). Por lo tanto, se traslada el atributo Nombre término a la entidad Términos, convirtiéndola en entidad débil:



Obsérvese la participación total de la entidad débil Términos en Pertenece.

c)

Entidades fuertes:

Idiomas(Nombre idioma)

Nombres de términos(Nombre término)

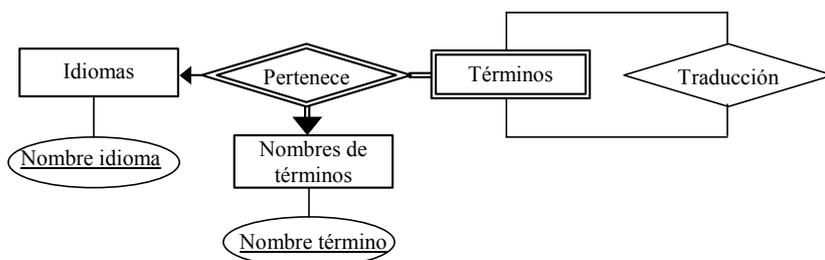
Entidades débiles:

Términos(Nombre idioma, Nombre término)

Relaciones:

Traducción(Nombre idioma1, Nombre término1, Nombre idioma2, Nombre término2)

Otra alternativa sería convertir la relación Pertenece en una relación ternaria como se muestra a continuación:



Obsérvese que se debe añadir la restricción de participación total entre Pertenece y Nombres de términos para evitar que al borrar un idioma queden nombres de términos sin asignar a ningún idioma. Sin embargo, este esquema no ofrece ninguna ventaja con respecto al anterior.

- 3) (3,3 puntos) Una compañía de seguros de automóviles desea informatizar la gestión de todas sus operaciones, para lo cual quiere crear una base de datos que recoja el funcionamiento de su negocio y que refleje la siguiente especificación de requisitos:

El elemento fundamental de información es la póliza, la cual se identifica mediante un número único, tiene un tipo de seguro (a todo riesgo, a terceros, etc.), un importe de cobertura máxima, importe de la póliza, un estatus (alta, baja, suspensión, etc.) y una fecha de contratación. La póliza pertenece a un único cliente (un cliente puede tener varias pólizas) y referencia a un único vehículo, y cada vehículo sólo puede tener una póliza.

Los clientes se identifican mediante su NIF, además se quiere guardar su nombre y apellidos, teléfono, fecha de nacimiento, fecha de obtención del permiso de conducir y su dirección completa (calle, número, código postal, localidad, provincia)

De los vehículos es necesario conocer su marca, modelo, matrícula, número de chasis, potencia, color y año de fabricación. Además un vehículo puede tener varios extras (alarma, auto-radio, etc.).

Una póliza puede tener una serie de personas autorizadas, de las cuales se quiere tener su NIF, nombre y apellidos, fecha de nacimiento y relación con el cliente. Un autorizado sólo tendrá relación con un cliente.

Cuando se produce un siniestro se crea una parte de accidente (identificado por un número de siniestro) donde se recoge la información del siniestro: datos de la póliza del cliente, datos del conductor (el conductor sólo puede ser un cliente o un autorizado), fecha del siniestro, nombre y número del perito de la agencia que ha autorizado la reparación, datos del taller donde se ha reparado el vehículo y fecha e importe de la reparación.

Si el accidente es con otro vehículo, no es necesario guardar información del vehículo contrario.

Además, sobre la base de datos se quiere realizar las siguientes operaciones (1) Anualmente se incrementa el importe de la póliza en un 2% por cada siniestro que ha tenido y, si la antigüedad del carné del cliente sobrepasa los 10 años, se disminuye en un 1% por cada año que sobrepase los 10 años. (2) La cobertura máxima se incrementa en un 2% durante los 5 primeros años de la póliza, y se decrementa en un 3% si la póliza sobrepasa los 10 años.

(3) Mensualmente se saca un listado con el número de siniestros de cada cliente y (4) semanalmente un listado con los siniestros revisados por cada perito.

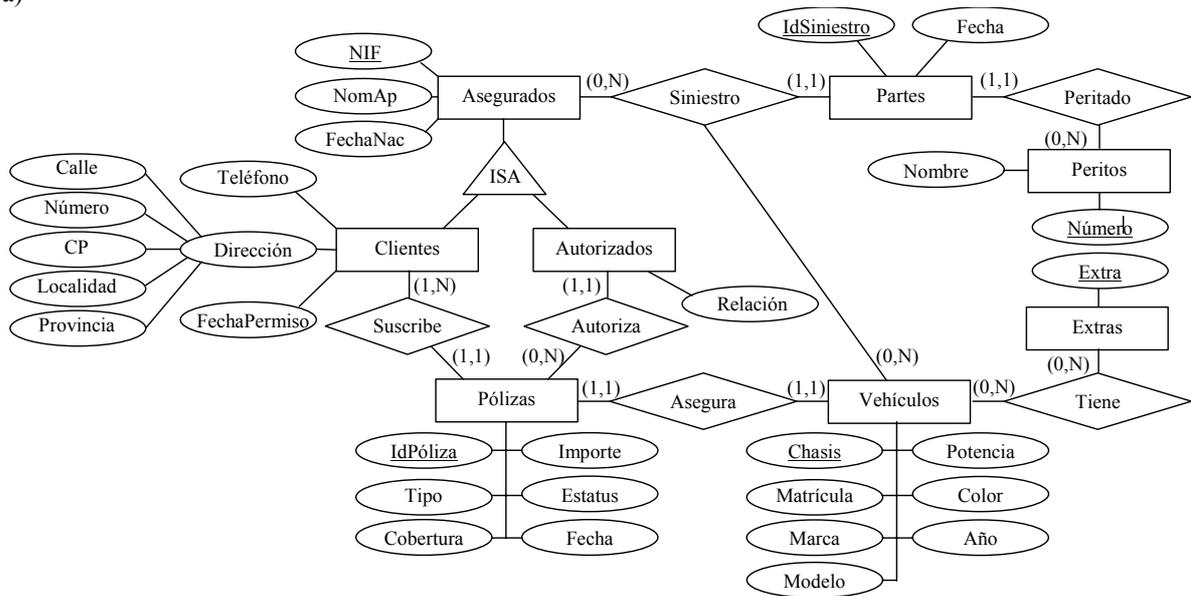
Se pide:

- Construir el esquema entidad/relación añadiendo el mínimo número de atributos, usando generalizaciones donde sea posible, imponiendo y explicando las restricciones de cardinalidad mínimo-máximo, sin considerar los costes de las operaciones pero asegurando que se puedan realizar.
- Traducir el esquema conceptual a un esquema lógico relacional expresando las restricciones de integridad referencial en notación algebraica.
- Plantear en SQL las operaciones (3) y (4).



Solución (50 min.):

a)



Entidades:

- **Asegurados.**
Es una generalización de Clientes y Autorizados. No lo es de Peritos porque sólo interesa el número de perito y hacerlo así afectaría a la relación Siniestro.
- **Clientes.**
- **Autorizados.**
- **Partes.**
- **Peritos.**
Se distingue como entidad en lugar de como atributo para conservar la restricción de que a un número de perito le corresponde un único nombre. En caso contrario habría que imponer la dependencia funcional {número de perito} → {nombre de perito}.
- **Pólizas.**
- **Vehículos.**
Se distingue como entidad por dos motivos: está implicada en Siniestro y en Tiene. No obstante, ambas se podrían eliminar: Siniestro relacionaría a Pólizas en lugar de a Vehículos y Extras sería un atributo multivalorado de Pólizas. Este atributo multivalorado tendría como dominio los posibles nombres de extras. Se deja mejor como entidad aparte porque permite definir dinámicamente este dominio (con los valores de la entidad Extras).
- **Extras.**

Relaciones:

- **Siniestro.** Debe ser ternaria para conocer el conductor y su vehículo implicado en un siniestro. No se podría conocer el vehículo si fuese binaria entre Asegurados y Partes.
- **Suscribe.** Debe ser binaria entre Clientes y Pólizas para conocer el tomador del seguro de cada póliza.
- **Autoriza.** Debe ser binaria entre Autorizados y Pólizas para conocer las personas autorizadas en cada póliza.
- **Asegura.** Binaria entre Pólizas y Vehículos.
- **Tiene.** Binaria entre Vehículos y Extras.
- **Peritado.** Binaria entre Partes y Peritos.

Cardinalidades:

- **Siniestro.**
Asegurados – (0,N) – Siniestro – Partes – Vehículos: Un asegurado (cliente o autorizado) puede no tener siniestros o tener más de uno con un vehículo.
Asegurados – Siniestro – (1,1) – Partes– Vehículos: Un parte sólo puede corresponder a un asegurado, y no puede haber partes sin relacionar.



- Asegurados – Partes – Siniestro – (0,N) – Vehículos: Un vehículo puede haber sufrido varios siniestros o ninguno.
- Suscribe.
Clientes – (1,N) – Suscribe – Pólizas: Un cliente puede suscribir una o más pólizas (para ser cliente al menos debe suscribir una).
Clientes – Suscribe – (1,1) – Pólizas: Cada póliza debe estar suscrita por algún cliente y sólo uno.
- Autoriza.
Pólizas – (0,N) – Autoriza – Autorizados: Una póliza puede autorizar a varias personas o a ninguna.
Pólizas – Autoriza – (1,1) – Autorizados: Cada autorizado debe estarlo en una y sólo una póliza. Esto asegura que sólo tenga relación con un único cliente.
- Asegura.
Pólizas – (1,1) – Asegura – Vehículos: Una póliza asegura a un y sólo un vehículo.
Pólizas – Asegura – (1,1) – Vehículos: Cada vehículo registrado en la compañía de seguros debe estar asegurado por una única póliza.
- Tiene.
Vehículos – (0,N) – Tiene – Extras: Un vehículo puede tener varios o ningún extra.
Vehículos – Tiene – (0,N) – Extras: Cada extra registrado puede estar asociado con varios o con ningún vehículo.
- Peritado.
Partes – (1,1) – Peritado – Peritos: Un parte está peritado por un y sólo un perito.
Partes – Peritado – (0,N) – Peritos: Un perito puede peritar uno o más partes.

b)

Fase de reestructuración:

1- Análisis de redundancias: no hay.

2- Eliminación de generalizaciones:

Sólo hay una: Asegurados Alternativas:

i) Plegar hacia el padre.

- Inconvenientes:
 - Aparecen muchos atributos NULL en los atributos de Clientes para cada entidad de Autorizados.
 - En el esquema se pierde información de las restricciones de pertenencia de las entidades Clientes y Autorizados en las relaciones Suscribe y Autoriza.
- Ventajas:
 - Se simplifican las tres entidades Asegurados, Clientes y Autorizados en una sola.

ii) Plegar hacia los hijos.

- Inconvenientes:
 - Hay que duplicar la relación ternaria Siniestro.
- Ventajas:
 - Se simplifican las tres entidades Asegurados, Clientes y Autorizados en una sola.

iii) Sustituir la generalización por relaciones binarias. Inconvenientes:

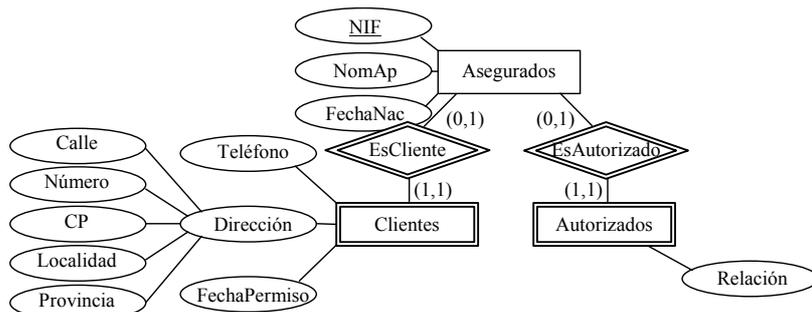
- Inconvenientes:
 - Aparecen dos nuevas relaciones. Sin embargo, debido a que la generalización es total (no hay asegurados que no sean clientes o autorizados), la cardinalidad es uno a uno tanto entre Asegurados y Clientes como entre Asegurados y Autorizados. Por lo tanto, podrían omitirse las dos tablas resultado de traducir las dos nuevas relaciones.
 - Aunque la cardinalidad es uno a uno, la participación mínima de asegurados en EsCliente y EsAutorizado (véase la figura a continuación) es 0, por lo que el esquema no puede asegurar la participación total. Para asegurarlo hay que añadir una nueva restricción al esquema:

$$\pi_{NIF}(Asegurados) \subseteq \pi_{NIF}(Clientes) \cup \pi_{NIF}(Autorizados)$$

Esta restricción no se puede expresar con restricciones de integridad referencial salvo que se añadan nuevas entidades.

- Ventajas:
 - No se pierde información sobre restricciones en el esquema resultante

Por lo tanto, se escoge esta última alternativa. El subesquema relativo a estas entidades queda:



3- División y mezcla de entidades y relaciones: No hay atributos multivalorados. No se dividen ni mezclan por consideraciones de coste (con respecto a las operaciones sólo se indica que se puedan realizar, no que se calcule su coste). Sólo hay un atributo compuesto (Dirección de Clientes). Como no se espera en general (sino en muy pocos casos) tener la misma dirección para varios clientes, en lugar de crear una relación se plegan los atributos de Dirección en Clientes.

Traducción a un esquema relacional:

Entidades fuertes:

- Asegurados(NIF, NomAp, FechaNac)
- Partes(IdSiniestro, Fecha)
- Peritos(Número, Nombre)
- Pólizas(IdPóliza, Tipo, Cobertura, Importe, Estatus, Fecha)
- Vehículos(Chasis, Matrícula, Marca, Modelo, Potencia, Color, Año). Clave candidata: {Chasis}
- Extras(Extra)

Entidades débiles:

- Clientes(NIF, Teléfono, Calle, Número, CP, Localidad, Provincia, FechaPermiso)
- Autorizados (NIF, Relación)

Relaciones:

- Siniestro(NIF, IdSiniestro, Matrícula)
- Suscribe(NIF, IdPóliza).
Esta tabla se podría eliminar añadiendo el atributo NIF a Pólizas.
- Autoriza(NIF, IdPóliza)
Esta tabla se podría eliminar añadiendo el atributo IdPóliza a Autorizados.
- Asegura(IdPóliza, Matrícula). Clave candidata: {Matrícula}
Esta tabla se podría eliminar añadiendo el atributo IdPóliza a Vehículos o Matrícula a Pólizas.
- Tiene(Matricula, Extras)
- Peritado(IdSiniestro, Número)
Esta tabla se podría eliminar añadiendo el atributo Número (de perito) a Partes.

Restricciones:

- Clave primaria (ya indicadas)
- Clave candidata (ya indicadas)
- Dependencias funcionales:
 - Relación Clientes:
{CP} → {Localidad, Provincia}
- Integridad referencial:
 - Entidades débiles:
 - Clientes:
 $\pi_{NIF}(Clientes) \subseteq \pi_{NIF}(Asegurados)$
 - Autorizados:
 $\pi_{NIF}(Autorizados) \subseteq \pi_{NIF}(Asegurados)$
 - Relaciones:
 - Siniestro:
 $\pi_{NIF}(Siniestro) \subseteq \pi_{NIF}(Asegurados)$
 $\pi_{IdSiniestro}(Siniestro) \subseteq \pi_{IdSiniestro}(Partes)$



$\{A\}^+ = \{A,D,B,E\}$ No es superclave
 $\{D\}^+ = \{D,B\}$ No es superclave

Se descompone R aplicando el algoritmo de descomposición a FNBC:

```
D={R}
while Q ∈ D, t.q. D no está en FNBC
  {encontrar X->Y de Q que viole FNBC
   reemplazar Q por Q-Y y X∪Y}
```

$AB \rightarrow E$
 $Q-Y=R1(A,B,C,D)$
 $X\cup Y=R2(A,B,E)$

Para comprobar si R1 está en FNBC:

- Determinar el cierre de F (F^+).
- Comprobar que para cada $X \rightarrow Y \in F^+$, X es superclave.

$\{A\}^+ = \{A,D,B\}$ No es superclave, $\{A \rightarrow BD\}$

Se descompone R1 con respecto a $\{A \rightarrow BD\}$

$Q-Y=R11(A,C)$
 $X\cup Y=R12(A,B,D)$
 Se ha terminado con R11 porque sólo tiene dos atributos.
 Se comprueba R12
 $\{A\}^+ = \{A,D,B\}$ Superclave
 $\{B\}^+ = \{B\}$
 $\{D\}^+ = \{B,D\}$ No es superclave $\{D \rightarrow B\}$

Se descompone R12 con respecto a $\{D \rightarrow B\}$

$Q-Y=R121(A,D)$
 $X\cup Y=R122(B,D)$
 Se ha terminado con R121 y R122 porque sólo tienen dos atributos.

Se comprueba R2(A,B,E)

$\{A\}^+ = \{A,D,B,E\}$ Superclave
 $\{B\}^+ = \{B\}$
 $\{E\}^+ = \{E\}$
 $\{B,E\}^+ = \{B,E\}$

Está en FNBC porque no hay DF cuyo antecedente no sea superclave (todos los que contengan A son superclave).

La solución es:

R11(A,C)
 R121(A,D)
 R122(B,D)
 R2(A,B,E)

b) ¿La transformación ha preservado las dependencias funcionales?

Se podría pensar que $A \rightarrow BD$ no se preserva porque no hay ninguna tabla que contenga estos tres atributos. Sin embargo, y dado que es una dependencia funcional que se puede descomponer (véase por ejemplo el cálculo del recubrimiento mínimo del apartado c), hay que comprobarlo.

Se aplica el algoritmo:

```
for each D.F. X → Y ∈ F
1. Z := {X}
```



```

while cambios en Z do
  for i:=1 to k do
2.      Z := Z ∪ ((Z∩Ri)+∩Ri)
Si Y ⊆ Z ⇒ X → Y ∈ F+
Si hay algún X → Y t.q. Y no es subcjto. de Z, significa que X → Y no ∈ G+
y, por tanto, no se conservan las D.F.

```

Para $AB \rightarrow E$:

No es necesario aplicar el algoritmo porque se preserva en R2 (tiene estos tres mismos argumentos).

Para $A \rightarrow BD$:

1. $Z = \{A\}$
- 2.R11(A,C) $Z = \{A\} \cup ((\{A\} \cap \{A,C\})^+ \cap \{A,C\}) = \{A\} \cup (\{A,D,B,E\} \cap \{A,C\}) = \{A\}$
- 2.R121(A,D) $Z = \{A\} \cup ((\{A\} \cap \{A,D\})^+ \cap \{A,D\}) = \{A\} \cup (\{A,D,B,E\} \cap \{A,D\}) = \{A,D\}$
3. R122(B,D) $Z = \{A,D\} \cup ((\{A,D\} \cap \{B,D\})^+ \cap \{B,D\}) = \{A,D\} \cup (\{B,D\} \cap \{B,D\}) = \{A,B,D\}$

Aquí ya sabemos que la dependencia funcional se preserva porque $\{BD\}$ es subconjunto de Z, no es necesario continuar con R2.

Es lógico que esta dependencia funcional se preserve porque se puede descomponer en $A \rightarrow B$ y $A \rightarrow D$. La primera se preserva en R2 y la segunda en R121. El algoritmo comprueba que $A \rightarrow D$ se preserva en R121 y que $A \rightarrow B$ también se preserva, pero no en R2 (de hecho no ha sido necesario usarla en el algoritmo), sino que se preserva $A \rightarrow D$ (en R121) y $D \rightarrow B$ (en R122), que son equivalentes por el axioma de transitividad a $A \rightarrow B$.

Para $D \rightarrow B$:

No es necesario aplicar el algoritmo porque se preserva en R122 (tiene estos dos mismos argumentos).

c) Descomponer R para obtener un conjunto de esquemas en 3FN.

En primer lugar se comprueba si se encuentra ya en 3FN:

Para toda $X \rightarrow Y$, o bien X es superclave o Y contiene algún atributo que pertenece a una clave candidata.

No hay claves candidatas de un atributo.

Si hay de dos, deberían contener a C y A (porque no se encuentran en ningún consecuente)

$\{AC\}^+ = \{A,B,C,D,E\}$ Clave candidata (obviamente, no hay más)

Para $AB \rightarrow E$, AB no es superclave y E no es parte de ninguna clave candidata. Por lo tanto, R no está en 3FN.

Se aplica el algoritmo de descomposición en 3FN:

$D = \{R\}$

1. Encontrar un recubrimiento mínimo T de F
2. for each $X \rightarrow Y \in T$, crear en D un esquema R_i con $\{X \cup \{A_1\} \cup \dots \cup \{A_k\}\}$ si $R_i \not\subseteq R_j \in D$, dadas las D.F. $X \rightarrow \{A_1\}, \dots, X \rightarrow \{A_k\}$
3. Si ninguno de los esquemas contiene una clave candidata de R, se crea uno nuevo.

1. Para encontrar T:

$A \rightarrow BD$ se descompone en $A \rightarrow B$ y $A \rightarrow D$

Para $AB \rightarrow E$ comprobamos si se puede eliminar B:

$\{AB\}^+ = \{ABDE\}$

$\{A\}^+ = \{ABDE\}$ Se conserva el cierre, por lo que nos quedamos con $\{A \rightarrow E, A \rightarrow B, A \rightarrow D, D \rightarrow B\}$

Finalmente, comprobamos si se puede eliminar alguna de las dependencias funcionales:

$A \rightarrow E$ no se puede eliminar porque ninguna otra dependencia funcional contiene a E.

Vemos si $A \rightarrow B$ se puede eliminar:

$\{A\}^+_{\{A \rightarrow E, A \rightarrow D, D \rightarrow B\}} = \{A,B,D,E\}$

Por lo tanto, se puede eliminar.

Las últimas dos dependencias funcionales no se pueden eliminar por el mismo motivo que la primera.



Por lo tanto, $T = \{ A \rightarrow E, A \rightarrow D, D \rightarrow B \}$

2. $A \rightarrow E: \{A,D,E\} \Rightarrow R1(A,D,E)$

2. $D \rightarrow B: \{B,D\} \Rightarrow R2(B,D)$

3. AC es clave candidata y no se encuentra en las anteriores; por lo tanto, se añade $R3(A,C)$

Por lo tanto, la descomposición en 3FN queda:

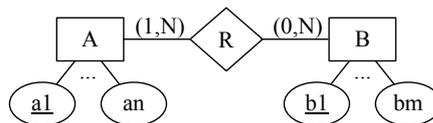
$R1(A,D,E)$

$R2(B,D)$

$R3(A,C)$

- 5) (2 puntos) Dado el esquema entidad-relación, las tablas de volúmenes de datos y de frecuencias y las operaciones de la siguiente figura se pide:
- Traducirlo a un esquema relacional con tres tablas según el diseño sistemático clásico de bases de datos.
 - Calcular el coste global de OP1 y OP2 suponiendo que los accesos en escritura tienen un coste doble que el de lectura y que el coste de acceso a las tuplas es constante (independiente del número de atributos).
 - Proponer un nuevo esquema relacional en el que se reduzca al mínimo el coste global de estas operaciones.
 - Calcular el espacio desperdiciado en (c) debido a las redundancias suponiendo que cada atributo ocupa lo mismo.

Concepto	Volumen
A	1.000
B	10.000
R	5.000



Operación	Frecuencia	Descripción
OP1	100	Listar los atributos a1, ..., an, b1, ..., bm de las entidades A y B relacionadas
OP2	10	Insertar relación entre A y B

Solución (30 min):

a)

$A(a_1, a_2, \dots, a_n)$

$B(b_1, b_2, \dots, b_m)$

$R(a_1, b_1)$

b)

Tabla de accesos

OP1			
Concepto	Accesos	Tipo	Descripción
R	5.000	L	
A	5.000	L	
B	5.000	L	
OP2			
Concepto	Accesos	Tipo	
R	1	E	
A	1	L	Para comprobar la restricción de integridad referencial
B	1	L	Para comprobar la restricción de integridad referencial

Coste OP1 = $(5000+5000+5000)*100 = 1.500.000$

Coste OP2 = $(1*2+1+1)*10 = 40$

Coste global = 1.500.040

c)

En este segundo apartado se admiten dos posibilidades: si se permite aumentar el número de tablas del esquema o si no.

i) Se permite aumentar el número de tablas.



En este caso y, dado que el coste de los accesos por tupla se considera constante (independiente de su número de columnas), es evidente que se puede simplemente añadir una tabla R2 con las mismas filas que R pero con las columnas a_2, \dots, a_n y b_2, \dots, b_m que provienen de A y B respectivamente, y con los valores adecuados relacionados por R. De esta forma:

Tabla de accesos

OP1			
Concepto	Accesos	Tipo	Descripción
R2	5.000	L	
OP2			
Concepto	Accesos	Tipo	
R	1	E	
A	1	L	Para comprobar la restricción de integridad referencial de R
B	1	L	Para comprobar la restricción de integridad referencial de R
R2	1	E	
A	1	L	Para comprobar la restricción de integridad referencial de R2
B	1	L	Para comprobar la restricción de integridad referencial de R2

$$\text{Coste OP1} = (5000) * 100 = 500.000$$

$$\text{Coste OP2} = (1 * 2 + 1 + 1 + 1 * 2 + 1 + 1) * 10 = 80$$

$$\text{Coste global} = 500.080$$

Aunque se ha duplicado el coste de la operación OP2, es despreciable con respecto al coste de la operación OP1.

ii) En este segundo caso no se admite aumentar el número de tablas, con lo cual aparecen las siguientes posibilidades de simplificación:

- c1) Fundir las tablas A y R
- c2) Fundir las tablas B y R
- c3) Fundir las tablas A, B y R

c1) Fundir las tablas A y R:

A($\underline{a_1}$, $\underline{b_1}$, a_2 , ..., a_n)

B($\underline{b_1}$, b_2 , ..., b_n)

El nuevo volumen de A sería 5.000 para representar R

Tabla de accesos

OP1			
Concepto	Accesos	Tipo	Descripción
A	5.000	L	
B	5.000	L	
OP2			
Concepto	Accesos	Tipo	
A	1	E	
B	1	L	Para comprobar la restricción de integridad referencial

$$\text{Coste OP1} = (5.000 + 5.000) * 100 = 1.000.000$$

$$\text{Coste OP2} = (1 * 2 + 1) * 10 = 30$$

$$\text{Coste global} = 1.000.030$$

Con esta alternativa se reduce el coste en un tercio aproximadamente y tiene el inconveniente:

- Se debe imponer la dependencia funcional: $a_1 \rightarrow a_2, \dots, a_n$.

c2) Fundir las tablas B y R:

A($\underline{a_1}$, a_2 , ..., a_n)

B(a_1 , b_1 , b_2 , ..., b_n)



Para calcular el nuevo volumen de B, dado que no tiene participación total en R, tenemos en el peor caso que cada A se relaciona con el mismo B (se necesitan 1.000 – 1 filas adicionales en B para representar R) y en el mejor que se relaciona con B's diferentes (no se necesita ninguna fila adicional).

Volumen de B = $10.000 + (1.000 - 1 + 0)/2 = 10.500$ aprox.

Tabla de accesos

OP1			
Concepto	Accesos	Tipo	Descripción
A	5.000	L	
B	10.500	L	
OP2			
Concepto	Accesos	Tipo	
A	1	L	Para comprobar la restricción de integridad referencial
B	1	E	

Coste OP1 = $(5.000+10.500)*100 = 1.550.000$

Coste OP2 = $(1+1*2)*10 = 30$

Coste global = 1.550.030

Con esta alternativa se pierde algo de eficiencia y sus inconvenientes son:

- Pueden aparecer valores nulos en el atributo a1 de B (no todos los B están relacionados con algún A).
- Por lo tanto, no se puede imponer restricción de clave primaria.
- Se debe imponer la dependencia funcional: $b1 \rightarrow b2, \dots, bm$.

c3) Fundir las tablas A, B y R:

$R(a1, \dots, an, b1, \dots, bm)$

El nuevo volumen de R es $1.000*5$ (cada entidad de A está relacionada en media con 5 de B) + $((10.000-1)+(10.000-1.000))/2$ (En el peor de los casos cada entidad de A está relacionada con una sola de B y en el mejor cada entidad de A está relacionada con una distinta de B) = $5.000 + 9.500 = 14.500$ aprox.

Tabla de accesos

OP1			
Concepto	Accesos	Tipo	Descripción
R	14.500	L	
OP2			
Concepto	Accesos	Tipo	
R	1	E	

Coste OP1 = $(14.500)*100 = 1.450.000$

Coste OP2 = $(1*2)*10 = 20$

Coste global = 1.450.020

Esta alternativa es un poco más eficaz que la original pero los inconvenientes son:

- Pueden aparecer valores nulos en el atributo a1 de R (no todos los B están relacionados con algún A).
- Por lo tanto, no se puede imponer restricción de clave primaria.
- Se deben imponer las dependencias funcionales: $a1 \rightarrow a2, \dots, an$ y $b1 \rightarrow b2, \dots, bm$.

Por lo tanto, la mejor alternativa desde el punto de vista del rendimiento es la c1)

d)

Según los dos supuestos del apartado c):

i) Se añade la tabla R2.

El espacio desperdiciado es el volumen ocupado por la tabla R2, es decir:

Volumen(R2) = $5.000*(n+m)$

ii)

El espacio desperdiciado en c1) se calcula:

Volumen original (VO) = $1.000*n$ (de A) + $10.000*m$ (de B) + $5.000*2$ (de R)



Volumen de c1) (VC1) = 5.000*n (de A) + 10.000*m (de B)

Espacio desperdiciado = VC1 - VO = 4.000*n - 10.000

que sólo es positivo si $n \geq 3$. Es decir, si $n=2$, el esquema no sólo es más eficaz sino que también ahorra espacio (2.000 unidades).

- 6) (1 punto) Dada la base de datos de programadores, analistas y proyectos definida por el esquema relacional:

Programadores(DNI, Nombre, Dirección, Teléfono)

Analistas(DNI, Nombre, Dirección, Teléfono)

Distribución(CódigoPr, DNIEmp, Horas)

Proyectos(Código, Descripción, DNIDir)

Se pide escribir consultas en álgebra relacional, cálculo relacional de tuplas y cálculo relacional de dominios para obtener la siguiente información:

- Nombre de los programadores que han participado en el proyecto de código P1.
- Nombres de los programadores y analistas que participan en algún proyecto dirigido por el analista de nombre A1.

Solución (15 min):

a)

AR: $\Pi_{\text{Nombre}}(\sigma_{\text{CódigoPr}='P1'}(\text{Programadores} \bowtie_{\text{DNI}=\text{DNIEmp}} \text{Distribución}))$

CRT: $\{t \mid t \in \text{Programadores} \wedge (\exists u \in \text{Distribución} (t[\text{DNI}] = u[\text{DNIEmp}] \wedge u[\text{CodigoPr}] = 'P1'))\}$

CRD: $\{ \langle n \rangle \mid \exists \text{dni}, d, t (\langle \text{dni}, n, d, t \rangle \in \text{Programadores} \wedge \exists c, h (\langle c, \text{dni}, h \rangle \in \text{Distribución})) \}$

b)

AR: $\Pi_{\text{Nombre}}(\sigma_{\text{NombreAnalista}='A1'}(((\text{Programadores} \cup \text{Analistas}) \bowtie_{\text{DNI}=\text{DNIEmp}} \text{Distribución}) \bowtie_{\text{Código}=\text{CódigoPr}} \text{Proyectos}) \bowtie_{\text{DNIDir}=\text{DNIAnalista}} \rho_{\text{Analistas}(\text{DNIAnalista}, \text{NombreAnalista}, \text{Dirección}, \text{Teléfono})}(\text{Analistas}))$

CRT: $\{t \mid t \in \text{Programadores} \wedge (\exists d \in \text{Distribución} (t[\text{DNI}] = d[\text{DNIEmp}] \wedge (\exists p \in \text{Proyectos} (d[\text{CódigoPr}] = p[\text{Código}]) \wedge (\exists a \in \text{Analistas} (a[\text{DNI}] = p[\text{DNIEmp}] \wedge a[\text{Nombre}] = 'A1'))))) \vee t \in \text{Analistas} \wedge \dots \}$

CRD: $\{ \langle n \rangle \mid \exists \text{dni}, d, t (\langle \text{dni}, n, d, t \rangle \in \text{Programadores} \wedge \exists c, h (\langle c, \text{dni}, h \rangle \in \text{Distribución}) \wedge \exists d, \text{dir} (\langle c, d, \text{dir} \rangle \in \text{Proyectos}) \wedge \exists \text{ddir}, \text{tdir} (\langle \text{dir}, 'A1', \text{ddir}, \text{tdir} \rangle \in \text{Analistas})) \vee \exists \text{dni}, d, t (\langle \text{dni}, n, d, t \rangle \in \text{Analistas} \dots) \}$