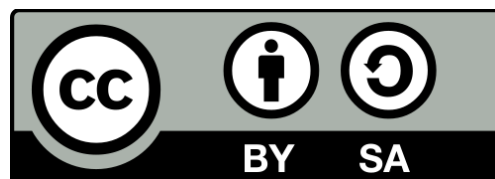


Fundamentos de las Bases de datos

Ingeniería en Informática

Ingeniería Multimedia



SB soluciones por [BDgite](#) se encuentra bajo una [Licencia Creative Commons Atribución-CompartirIgual 3.0 Unported](#).

Basada en una obra en <http://fbddocs.dlsi.ua.es>.

Permisos que vayan más allá de lo cubierto por esta licencia pueden encontrarse en <http://fbddocs.dlsi.ua.es/autores>.



BDgite (GITE-11014-UA)

lsi | Departamento
de Lenguajes
y Sistemas
Informáticos



Universitat d'Alacant
Universidad de Alicante

SOLUCIONES SEGUNDA HORA

SB02

- 1) Las claves candidatas no pueden almacenar NULOS porque... **el nulo es ignorancia, no sé si tiene valor y, si lo tiene, no puedo asegurar cuál es y, por tanto, podría ser un duplicado en un estado de base de datos concreto.**
- 2) Oracle 10g, MySQL, etc., al ser sistemas de gestión de bases de datos (SGBD) relacionales
 - a) jamás permitirán tablas sin clave primaria **FALSO**
 - b) si definimos una tabla sin clave primaria, utilizarán automáticamente todas sus columnas para definir la clave. **FALSO**
Estos son productos comerciales y, como tales, implementaciones del modelo relacional que dan servicio a quien los adquiera. A veces es necesario definir tablas sin clave primaria, mayormente por cuestiones de mantenimiento y desarrollo de una base de datos. Podemos resumir que "una cosa es la teoría y otra el trabajo diario en base de datos".
En cualquier caso, y finalizadas esas tareas de mantenimiento o desarrollo, si queremos una base de datos relacional que vigile las propiedades de integridad de nuestro sistema de información, es poco aconsejable permitir tablas sin clave primaria.
- 3) Sea **tablaX (A, B, C, D)**
 Si definimos **Clave Primaria (A, B, C)** entonces ¿podrían ser claves alternativas los siguientes grupos de atributos?
 - a) (A) **no**
 - b) (B) **no**
 - c) (C) **no**
 - d) (D) **sí**
 - e) (A, B) **no**
 - f) (A, C) **no**
 - g) (A, D) **sí**
 - h) (B, C) **no**
 - i) (B, D) **sí**
 - j) (C, D) **sí**
 - k) (A, B, D) **sí**
 - l) (A, C, D) **sí**
 - m) (B, C, D) **sí**
 - n) (A, B, C, D) **no**
- 4) Sea una tabla **R(A, B, C, D)**. Suponiendo que la propiedad de identificación se va a cumplir elijamos lo que elijamos y que quisiéramos obtener la máxima cantidad de claves candidatas, esto es, elegir conjuntos de atributos de tal forma que se utilicen todas las columnas y que ninguno fuera subconjunto de otro, ¿hasta cuántas CC podría llegar a tener R?
Si empezáramos eligiendo (A,B,C) como clave primaria solo podríamos elegir (D) como clave alternativa, es decir, podríamos llegar a tener 2 claves candidatas.
Si elegimos (A) como clave primaria, también podríamos definir (B), (C) y (D) como alternativas, es decir, 4 claves candidatas en la tabla.
La combinación de claves candidatas más poblada, en esta tabla concreta, es aquella se forma agrupando 2 columnas, lo que nos da una clave primaria y 5 alternativas: (A,B), (A,C), (A,D), (B,C), (B,D), (C,D).
- 5) Dado el esquema siguiente, ¿cuáles de las siguientes operaciones no llegarán a realizarse (el motor de la base de datos nos devolverá un error de ejecución)?

T1(a,b,c) CP(a)

- a) insertar(1,10,100)
- ~~b) insertar(NULO,20,NULO)~~
- c) insertar(2,20,NULL)
- d) insertar(3,NULL,300)
- ~~e) insertar(2,NULO,NULO)~~
- ~~f) insertar(2,20,NULO)~~
- ~~g) insertar(1,20,200)~~

- 6) Dado el esquema siguiente, ¿cuáles de las siguientes operaciones no llegarán a realizarse (el motor de la base de datos nos devolverá un error de ejecución)?

T1(a,b,c) CP(b,c)

- a) insertar(1,10,100)
- ~~b) insertar(NULO,20,NULO)~~
- ~~c) insertar(2,20,NULL)~~
- ~~d) insertar(3,NULL,300)~~
- ~~e) insertar(2,NULO,NULO)~~
- ~~f) insertar(2,20,NULO)~~
- g) insertar(1,10,200)
- h) insertar(1,20,200)
- i) insertar(1,20,100)
- ~~j) insertar(2,10,100)~~

- 7) Dado el esquema y ejecutándose la siguiente secuencia de órdenes, ¿cuáles de las siguientes operaciones no llegarán a realizarse (el motor de la base de datos nos devolverá un error de ejecución)?

T1(a,b,c) CP(b,c)

- a) insertar(1,10,100)
- b) insertar(2,20,200)
- c) modificar(1,10,100) a (1,20,100)
- ~~d) modificar(1,20,100) a (1,20,200)~~
- e) insertar(1,10,10)
- ~~f) insertar(2,20,100)~~
- g) eliminar(1,20,100)
- h) insertar(2,20,100)
- i) insertar(3,30,300)
- ~~j) modificar(2,20,100) a (1,20,200)~~
- k) modificar(3,30,300) a (2,30,300)

SB03

- 1) ¿Cuáles de los siguientes esquemas es posible y cuáles no? (se supone que los dominios están correctamente definidos y son coherentes con las definiciones de claves; cada apartado es un esquema diferente).

- a) T1(a,b,c,d) CP(a,b,c,d)
- b) T1(a,b,c,d) CP(a,b,c)
- c) T1(a,b,c,d) CP(a) CAlt(b)
- ~~d) T1(a,b,c,d) CP(b) CAlt(a), CAlt(b,c)~~
- ~~e) T1(a,b,c,d) CP(a,b,c,d) CAlt(a,c,d)~~
- f) T1(a,b,c,d) CP(a,b,c) CAlt(a,c,d)
- g) T1(a,b,c,d) CP(a,b) CAlt(a,c) CAlt(a,d) CAlt(b,c) CAlt(b,d) CAlt(d,c)
- h) T1(a,b,c,d) CP(a) CAj(b) → T2
- i) T1(a,b,c,d) CP(a) CAj(a) → T2
- j) T1(a,b,c,d) CP(a) CAj(a,b,c,d) → T2
- k) T1(a,b,c,d) CP(a,b) CAj(b) → T2
- l) T1(a,b,c,d) CP(a) CAlt(b,c) CAj(b,a) → T2 CAj(b) → T3 CAj(a,c) → T4

- 2) Dado el esquema siguiente, ¿cuáles de las siguientes operaciones son "ilegales"? (Las órdenes se ejecutan en secuencia y dependen de las anteriores; se supone que los dominios están correctamente definidos y son coherentes con las definiciones de claves).

T1(a,b,c) CP(a) T2(a,d,e) CP(d) CAj(a) → T1

- a) insertar en T1(1,10,100)
- ~~b) insertar en T1(NULO,20,NULO)~~
- c) insertar en T1(2,20,NULO)
- d) insertar en T1(3,NULO,300)
- ~~e) insertar en T2(2,NULO,NULO)~~
- f) insertar en T2(2,20,NULO)
- ~~g) insertar en T1(1,20,200)~~
- ~~h) insertar en T2(4,10,100)~~
- i) insertar en T2(2,30,230)
- ~~j) modificar T1(1,10,100) a (2,10,100)~~
- k) modificar T1(1,10,100) a (5,10,100)
- l) modificar T2(2,20,NULO) a (2,20,220)
- m) modificar T2(2,20,220) a (5,20,220)
- n) modificar T2(5,20,220) a (2,10,100)
- ~~o) modificar T1(2,20,NULO) a (6,60,600)~~
- p) modificar T1(3,NULO,300) a (7,70,700)
- q) modificar T2(2,10,100) a (7,10,100)
- r) modificar T2(2,30,230) a (7,30,230)
- s) modificar T1(2,20,NULO) a (6,60,600)
- t) borrar T2(7,30,230)
- ~~u) borrar T1(7,70,700)~~
- v) borrar T1(5,10,100)
- w) borrar T2(7,10,100)
- x) borrar T1(7,70,700)
- y) borrar T1(6,60,600)

- 3) Dado el esquema siguiente, ¿cuáles de las siguientes operaciones son "ilegales"? (Las órdenes se ejecutan en secuencia y dependen de las anteriores; se supone que los dominios están correctamente definidos y son coherentes con las definiciones de claves).

T1(a,b,c) CP(a,b) T2(a,d,e) CP(d) CAj(a,d) → T1

- ~~a) insertar en T1(1,NULO,100)~~
- ~~b) insertar en T1(NULO,10,100)~~
- ~~c) insertar en T1(NULO,NULO,100)~~
- d) insertar en T1(1,10,110)
- e) insertar en T2(1,10,1000)
- ~~f) insertar en T2(1,10,2000)~~
- g) insertar en T1(1,20,120)
- h) insertar en T1(2,20,220)
- ~~i) insertar en T2(2,10,2100)~~
- ~~j) modificar T2(1,10,1000) a (1,NULO,NULO)~~
- ~~k) modificar T1(1,10,110) a (1,30,130)~~
- ~~l) modificar T2(1,10,1000) a (1,NULO,NULO)~~
- ~~m) borrar T1(1,10,110)~~
- n) modificar T2(1,10,1000) a (1,20,1200)
- o) borrar T1(1,10,110)
- p) borrar T1(2,20,220)
- q) borrar T2(1,20,1200)
- r) borrar T1(1,20,120)

SB04

1)

- a) Card(alumno, matrícula) = (0, 1) Card(asignatura, matrícula) = (0, N)
- b) Card(alumno, matrícula) = (0, N) Card(asignatura, matrícula) = (0, 1)
- c) Card(alumno, matrícula) = (1, 1) Card(asignatura, matrícula) = (0, N)
- d) Card(alumno, matrícula) = (0, 1) Card(asignatura, matrícula) = (1, 1)
- e) Card(alumno, matrícula) = (0, N) Card(asignatura, matrícula) = (0, N)
- f) Card(alumno, matrícula) = (0, 1) Card(asignatura, matrícula) = (0, 1)
- g) Card(alumno, matrícula) = (1, 1) Card(asignatura, matrícula) = (1, 1)

(para dar esta solución es necesaria más información, alguien debe decirnos que se trata de una relación ya que no hay claves ajenas que nos permitan suponerlo)

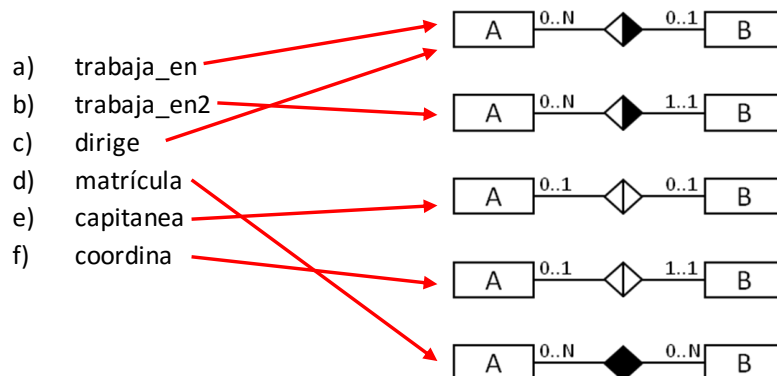
SB05

1)

- a) Card(DEPARTAMENTO, trabaja-en) = (0, N) Card(EMPLEADO, trabaja-en) = (0, 1)
- b) Card(DEPARTAMENTO, trabaja-en2) = (0, N) Card(EMPLEADO, trabaja-en2) = (1, 1)
- c) Card(DEPARTAMENTO, dirige) = (0, 1) Card(EMPLEADO, dirige) = (0, N)
- d) Card(ALUMNO, matrícula) = (0, N) Card(ASIGNATURA, matrícula) = (0, N)
- e) Card(BARCO, capitanea) = (0, 1) Card(CAPITÁN, capitanea) = (0, 1)
- f) Card(ASIGNATURA, coordina) = (1, 1) Card(PROFESOR, coordina) = (0, 1)
- g) Card(DEPARTAMENTO, trabaja-en) = (0, N) Card(EMPLEADO, trabaja-en) = (1, 1)
- Card(DEPARTAMENTO, dirige) = (0, 1) Card(EMPLEADO, dirige) = (0, N)

(si se elige EMPLEADO.dpto para "trabaja_en" y DEPARTAMENTO.jefe para "dirige")

2)



3)

2 tablas que se relacionan, con 1 clave ajena que se aloja en una de ellas.

2 tablas que se relacionan, con 1 clave ajena que se aloja en una de ellas y es de VNN.

2+1 tablas que se relacionan, con 2 claves ajenas que se alojan en la tercera tabla componiendo la clave primaria.

2+1 tablas que se relacionan, con 2 claves ajenas que se alojan en la tercera tabla, una como clave primaria y la otra como alternativa.

2 tablas que se relacionan, con 1 clave ajena que se aloja en una de ellas y es clave alternativa.

1 única tabla, con 1 clave alternativa (aparte, claro está, de la clave primaria).

4)

<p>a)</p> <p>PROGRAMA(siglas, nombre, inv)</p> <p>CP(siglas)</p> <p>CAj(inv) → INVITADO</p> <p>VNN(inv)</p> <p>INVITADO(dni, nombre)</p> <p>CP(dni)</p>	<p>b)</p> <p>PROGRAMA(siglas, nombre)</p> <p>CP(siglas)</p> <p>INVITADO(dni, nombre, prog)</p> <p>CP(dni)</p> <p>CAj(prog) → PROGRAMA</p> <p>VNN(prog)</p>
<p>c)</p> <p>PROGRAMA(siglas, nombre)</p> <p>CP(siglas)</p> <p>INVITADO(dni, nombre)</p> <p>CP(dni)</p> <p>COTILLEA(inv, prog)</p> <p>CP(inv, prog)</p> <p>CAj(prog) → PROGRAMA</p> <p>CAj(inv) → INVITADO</p>	<p>d)</p> <p>PROGRAMA(siglas, nombre)</p> <p>CP(siglas)</p> <p>INVITADO(dni, nombre, prog)</p> <p>CP(dni)</p> <p>CAj(prog) → PROGRAMA</p> <p>INSULTA(inv, prog)</p> <p>CP(inv)</p> <p>CAlt(prog)</p> <p>CAj(prog) → PROGRAMA</p> <p>CAj(inv) → INVITADO</p>
<p>e)</p> <p>PROGRAMA(siglas, nombre)</p> <p>CP(siglas)</p> <p>INVITADO(dni, nombre, prog, prog2)</p> <p>CP(dni)</p> <p>CAj(prog) → PROGRAMA</p> <p>CAlt(prog2)</p> <p>CAj(prog2) → PROGRAMA</p>	<p>f)</p> <p>PROGRAMA(siglas, nombre) CP(siglas)</p> <p>INVITADO(dni, nombre) CP(dni)</p> <p>GRITA(inv, prog)</p> <p>CP(inv, prog)</p> <p>CAj(prog) → PROGRAMA</p> <p>CAj(inv) → INVITADO</p> <p>LADRA(inv, prog)</p> <p>CP(inv, prog)</p> <p>CAj(prog) → PROGRAMA</p> <p>CAj(inv) → INVITADO</p>

SB06

1.

- I. BORRAR (UNIVERSIDAD.NUM=1) **ok**
- II. MODIFICAR (UNIVERSIDAD.NUM=2, 10) **ok**
- III. MODIFICAR (UNIVERSIDAD.NUM=3, 20) **ok**
- IV. BORRAR (UNIVERSIDAD.NUM=4) **ok**
- V. BORRAR (DEPARTAMENTO.COD=DLSI, DEPARTAMENTO.U=20) **ok**
- VI. BORRAR (PROFESOR.DNI=22444444) **ok**

Estado final de la BD

UNIVERSIDAD	
NUM	NOMBRE
10	POLITÉCNICA VALENCIA
20	JAUME I

PROFESOR			
DNI	NOMBRE	DPTO	U

DEPARTAMENTO	
COD	U
DLSI	10

2.

- I. BORRAR (UNIVERSIDAD.NUM=1) **rechazo**
- II. MODIFICAR (UNIVERSIDAD.NUM=2, 10) **rechazo**
- III. MODIFICAR (UNIVERSIDAD.NUM=3, 20) **rechazo**
- IV. BORRAR (UNIVERSIDAD.NUM=4) **ok**
- V. BORRAR (DEPARTAMENTO.COD=CCIA, DEPARTAMENTO.U=1) **ok**
- VI. BORRAR (PROFESOR.DNI=22444444) **ok**

Estado final de la BD

UNIVERSIDAD			
NUM	NOMBRE		
1	ALICANTE		
2	POLITÉCNICA VALENCIA		
3	JAUME I		

PROFESOR				DEPARTAMENTO	
DNI	NOMBRE	DPTO	U	COD	U
21333333	RENATA	DLSI	1	DLSI	1
23555555	ABUNDIO	DLSI	1	DLSI	2
13666666	SEGISMUNDO	null	null	DLSI	3
31777777	YENIPHER	null	null		

3.

- I. BORRAR (UNIVERSIDAD.NUM=1) **rechazo**
- II. MODIFICAR (UNIVERSIDAD.NUM=2, 10) **ok**
- III. MODIFICAR (UNIVERSIDAD.NUM=3, 20) **ok**
- IV. BORRAR (UNIVERSIDAD.NUM=4) **ok**
- V. BORRAR (DEPARTAMENTO.COD=CCIA, DEPARTAMENTO.U=1) **rechazo**
- VI. BORRAR (PROFESOR.DNI=22444444) **ok**

Estado final de la BD

UNIVERSIDAD			
NUM	NOMBRE		
1	ALICANTE		
10	POLITÉCNICA VALENCIA		
20	JAUME I		

PROFESOR				DEPARTAMENTO	
DNI	NOMBRE	DPTO	U	COD	U
21333333	RENATA	DLSI	1	DLSI	1
23555555	ABUNDIO	DLSI	1	DLSI	10
13666666	SEGISMUNDO	CCIA	1	DLSI	20
31777777	YENIPHER	CCIA	1	CCIA	1

SB07

1.
 - a) $\text{Card}(\text{FACTURA}, \text{detalle}) = (0, N)$ $\text{Card}(\text{LÍNEA}, \text{detalle}) = (1, 1)$; dependencia de identificador
 - b) $\text{Card}(\text{COCHE}, \text{es-un}) = (1, 1)$ $\text{Card}(\text{VEHÍCULO}, \text{es-un}) = (0, 1)$; generalización (o especialización)
 - c) $\text{Card}(\text{COCHE}, \text{alquila}) = (0, N)$ $\text{Card}(\text{CLIENTE}, \text{alquila}) = (0, N)$
2.
 - a) CAlt
 - b) una única CAj, no dentro CC
 - c) CP compuesta de 2 CAj's
 - d) CAj dentro de CP
 - e) CAj es la CP
3.
 - a) $\text{Card}(\text{ASIGNATURA}, \text{matrícula}) = (0, 1)$ $\text{Card}(\text{ALUMNO}, \text{matrícula}) = (0, 1)$
 - b) $\text{Card}(\text{POBLACIÓN}, \text{rige}) = (0, 1)$ $\text{Card}(\text{ALCALDE}, \text{rige}) = (1, 1)$
4.
 - a) $\text{Card}(\text{ALUMNOS}, \text{notas}) = (0, N)$ $\text{Card}(\text{ASIGNATURAS}, \text{notas}) = (0, N)$
 $\text{Card}(\text{PROFESOR}, \text{imparte}) = (0, N)$ $\text{Card}(\text{ASIGNATURAS}, \text{imparte}) = (0, N)$
 - b) $\text{Card}(\text{RÍO}, \text{desemboca}) = (0, 1)$ $\text{Card}(\text{MAR}, \text{desemboca}) = (0, N)$
 $\text{Card}(\text{PROVINCIA}, \text{está_en}) = (0, 1)$ $\text{Card}(\text{COMUNIDAD}, \text{está_en}) = (0, N)$
 $\text{Card}(\text{RÍO}, \text{pasar_por}) = (0, N)$ $\text{Card}(\text{PROVINCIA}, \text{pasar_por}) = (0, N)$

SB08

1.1 Fecha de los Partidos conocidos

PARTIDOS

1.2 partidos conocidos

PARTIDOS[fecha]

1.3 Fecha y equipos local y visitante de los partidos conocidos

- PARTIDOS[fecha, local, visitante]
- PARTIDOS

1.4 Fecha y equipos locales de los partidos del Hércules como visitante

- ~~PARTIDO [fecha, local] donde visitante = 'Hércules'~~
- PARTIDO donde visitante = 'Hércules' [fecha, local]

1.5 Fecha y equipos locales de los partidos del Hércules como visitante posteriores a diciembre de 2002

- PARTIDOS donde visitante = 'Hércules' [fecha, local] donde fecha > '31/12/2002'
- PARTIDOS donde visitante = 'Hércules' y fecha > '31/12/2002' [fecha, local]

2 ciudades que tienen periódico y equipo

- PERIODICO ∞ EQUIPO [CIUDAD]
- PERIODICO \times EQUIPO donde PERIODICO.ciudad=EQUIPO.ciudad[EQUIPO.CIUDAD]
- PERIODICO[CIUDAD] \cap (EQUIPO[CIUDAD])

3 ciudades que tienen periódico o equipo

PERIODICO[CIUDAD] \cup (EQUIPO[CIUDAD])

4 ciudades que tienen periódico pero no equipo

PERIODICO[CIUDAD] \sim (EQUIPO[CIUDAD])

5 dni y nombre de los periodistas sin estudios (estudios = "")

PERIODISTA donde ESTUDIOS="" [NOMBRE,DNI]

6 periódicos en los que ningún periodista tiene estudios

PERIODICO[PERIODICO] \sim (PERIODISTA donde ESTUDIOS<>"" [PERIODICO])

7 dni y nombre de los periodistas que asisten a los partidos del Hércules como local

PERIODISTA ∞ VADEGORRA donde LOCAL="Hércules" [DNI,NOMBRE]

8 dni y nombre de los periodistas que escriben en el Información

PERIODISTA donde PERIODICO="Información" [DNI,NOMBRE]

9 dni y nombre de los periodistas que escriben en el Información y asisten a los partidos del Hércules como local
 PERIODISTA ∞ VADEGORRA donde LOCAL="Hércules" y PERIODICO="Información" [DNI,NOMBRE]

10 dni y nombre de los periodistas que asisten a los partidos del Hércules
 PERIODISTA ∞ VADEGORRA ∞ PARTIDOS donde LOCAL="Hércules" O VISITANTE="Hércules" [DNI,NOMBRE]

11 Periódicos y equipos que son de la misma ciudad
 PERIODICO ∞ EQUIPO [PERIODICO,NOMBRE]

12 Periodistas simpatizantes del Hércules
 PERIODISTA donde EQUIPO="Hércules"

13 nombre de los periódicos que tienen periodistas simpatizantes del Hércules
 PERIODISTA donde EQUIPO="Hércules" [PERIODICO]

14 nombre de los periódicos que tienen periodistas que asisten a partidos del Hércules o del Elche
 VADEGORRA ∞ PARTIDOS donde LOCAL="Hércules" o VISITANTE="Hércules" O LOCAL="Elche" o VISITANTE="Elche" ∞ PERIODISTA [PERIODICO]

15 Partidos a los que asisten periodistas enemigos del equipo local
 VADEGORRA ∞ PARTIDOS ∞ PERIODISTA donde VISITANTE=EQUIPO [FECHA, LOCAL]

16 Nombre y ciudad de los equipos, y si la ciudad tiene periódico, mostrar su nombre
 EQUIPOS \bowtie PERIODICO [NOMBRE, CIUDAD, PERIODICO]

17 Nombre de los periodistas que simpatizan con equipos de ciudades diferentes a las de su periódico
 PERIODISTA \bowtie PERIODICO \times (EQUIPO REN[NOMBRE|EQUIPO, CIUDAD|CIUDAD_E]) donde CIUDAD \neq CIUDAD_E [NOMBRE]

18 Dni de los periodistas que han asistido a menos de 2 partidos del Ciudad de Murcia como local
 Define V1 alias para VADEGORRA Define V2 alias para VADEGORRA
 PERIODISTA[dni]

-
 (V1 \times V2
 donde V1.dni=V2.dni
 Y V1.local=V2.local Y V1.local='Ciudad de Murcia'
 Y V1.fecha \neq V2.fecha
 [V1.dni])

19 Ciudad que es la primera alfabéticamente
 Define E1 alias para EQUIPO Define E2 alias para EQUIPO
 Define P1 alias para PERIODICO Define P2 alias para PERIODICO
 E1[ciudad] \cup (P1[ciudad])

-
 (
 E1[ciudad] \cup (P1[ciudad])
 \times
 (E2[ciudad] \cup (P2[ciudad]))
 donde E1.ciudad > E2.ciudad
 [E1.ciudad]
)

20 Dni y nombre de los periodistas que no asisten a ningún partido del Elche
 PERIODISTA [DNI,NOMBRE] -
 (PERIODISTA \bowtie VADEGORRA \bowtie PARTIDOS donde LOCAL="Elche" o VISITANTE="Elche" [DNI,NOMBRE])

21 dni y nombre de los periodistas que asisten a todos los partidos del Hércules (usa división)
 VADEGORRA[DNI,FECHA,LOCAL]¹ \div (PARTIDOS donde LOCAL="Hércules" O VISITANTE="Hércules"
 [FECHA,LOCAL]) ∞ PERIODISTA [DNI,NOMBRE]

¹ En este caso no hace falta la proyección (véase el ejercicio 1.3)

22 dni y nombre de los periodistas que asisten a todos los partidos del Hércules y a ninguno del Elche (usa división)
VADEGORRA ÷ (PARTIDOS donde LOCAL="Hércules" O VISITANTE="Hércules" [FECHA,LOCAL]) ∞ PERIODISTA
[DNI,NOMBRE] ~ (PERIODISTA ∞ VADEGORRA ∞ PARTIDOS donde LOCAL="Elche" o VISITANTE="Elche"
[DNI,NOMBRE])

~~23 Todos los periodistas tienen equipo (equipo<>"")~~

~~PERIODISTAS donde equipo = "" [dni]~~

~~(si el resultado es "vacío de filas", se entiende como CIERTO)~~

SB09

SB10

1

- a) Card(MODELO, R1) = (0, 1) Card(AGENCIA, R1) = (0, N)
 Card(FOTÓGRAFO, R2) = (0, N) Card(AGENCIA, R2) = (0, N)
 Card(MODELO, R3) = (0, N) Card(COMPAÑÍA, R3) = (0, N)
- b) Card(perro, esDe) = (0,1) Card(proprietario, esDe)= (0,N)
 Card(proprietario, pertenecer)=(0,N) Card(asociación, pertenecer)=(0,N)
 Card(perro, adiestrar)=(0,N) Card(escuela, adiestrar)=(0,N)
- c) Card(PROVEEDORES, pedidos) = (0, N) Card(PIEZAS, pedidos) = (0, N)
 Card(PEDIDOS, parapry) = (1, 1) Card(PROYECTOS, parapry) = (0, N) (dependencia de identificador)
 Se podría haber emparejado de otra forma las tablas, alternándose unas y otras, lo importante es que nos debe salir una relación N:M y una dependencia de identificador.
- d) Card(CLIENTE, R1) = (0, 1) Card(EMPRESA, R1) = (0, N)
 Card(VEHÍCULO, R2) = (0, 1) Card(EMPRESA, R2) = (0, N)
 Card(CLIENTE, R3) = (0, N) Card(VEHÍCULO, R3) = (0, N)
- e) Card(ALUMNO, se_matrícula)= (0,N) Card(ASIGNATURA, se_matricula)=(0,N)
 Card(ALUMNO, recibe)=(0,N) Card(ASIGNATURA, recibe)=(0,N)
 La tabla RECIBE está mal definida porque una de las dos claves candidatas ha de ser errónea. Suponemos que elegimos al azar (no disponemos de suficiente información para tomar una decisión argumentada) la CP(dni, curso, letra, cod) para resolver el ejercicio.
 Con las cardinalidades puestas, nos queda aún una clave ajena en RECIBE, que solucionamos a continuación.
 Card(recibe, asigna)= (1,1) Card(GRUPO, asigna)= (0,N)

SB11

1

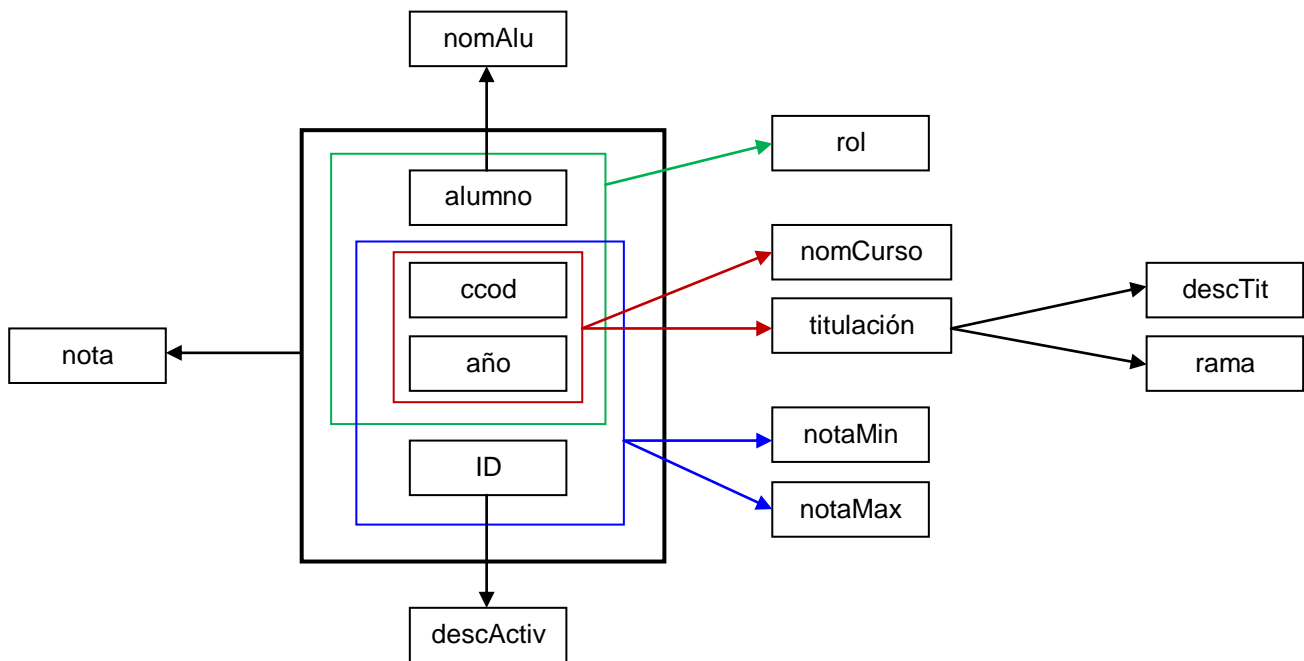
- Card(TITULACIÓN, pertenece)=(0,N) Card(CURSO, pertenece)= (1,1)
 Card(ALUMNO, es)= (1,1) Card(PERSONA, es)=(0,1)
 Card(PERSONA, participa)= (0,N) Card(CURSO, participa)= (0,N)
 Card(PARTICIPANTE, tiene)=(1,1) Card(ROL, tiene)= (0,N)
 Card(ACTIVIDAD, prevista)= (0,N) Card(CURSO, prevista)= (0,N)
 Card(PARTICIPANTE, califica)= (0,N) Card(PLANIFICACIÓN, califica)= (0,N)

2

- a) F
 b) V
 c) F
 d) V
 e) F
 f) F
 g) F
 h) F
 i) V
 j) F
 k) F

3

- a) opción **c**.
 b) opción **a**.
 c) opción **c**.



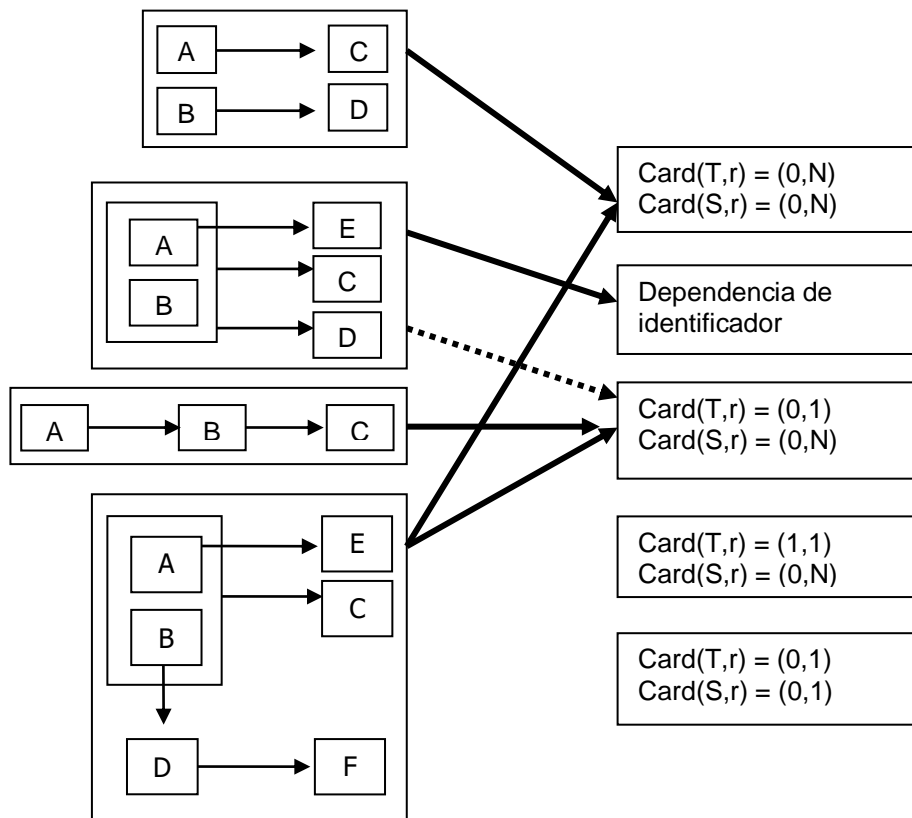
Precisamente, la tabla NOTAS presenta un solapamiento de atributos de la clave ajena que hace que todo el proceso sea muy extraño. De hecho, la idea es que un proceso de normalización es totalmente reversible, podríamos deshacer las proyecciones en nuevas tablas hasta agruparlo todo en una única tabla en 1FN. Sin embargo, las formas normales habituales no cubren todos los casos que el modelo relacional puede representar, por ejemplo, generalizaciones o algunas restricciones de existencia. NOTAS tampoco sale de un esquema E-R, ni tan siquiera de combinar tablas por relaciones 1:N o N:M. Es, simplemente, una tabla posible en MR, fruto de decisiones del diseñador trabajando directamente sobre el esquema lógico.

En resumen, no hay que tomarse este esquema como el resultado de un ejercicio de los vistos hasta ahora, pero sí asumimos que las tablas están en 3FN.

Como el ejercicio solo nos pide extraer las dependencias funcionales, lo haremos como si no supiéramos hacer nada más y de ahí el gráfico precedente. La mecánica es muy sencilla:

- los atributos no primos de una tabla dependen de forma completa de la clave primaria (en este caso no hay claves alternativas)
- las claves ajenas representan normalizaciones que, a su vez, indican dependencias funcionales a partir de las tablas a las que están apuntando.

5



La correspondencia punteada indica que se podrían asociar o no dependiendo de si se entiende que con la dependencia de identificador es suficiente, ya que implícitamente ya es una relación 1:N, o si se quiere reflejar y dar énfasis a esto último.

6

- a) CURSO (ccod:dcod, año:daño)
 CP (cod, año)
 CAj (ccod) → ACURSO

ACURSO (cod:dcod, nombre:cadena, titulación:dcod)
 CP (cod)
 CAj (titulación) → TITULACIÓN VNN(titulación)

Solo la tabla CURSO necesita ser modificada (se proyecta la nueva tabla ACURSO). El resto de tablas quedaría igual, en particular, TITULACIÓN. El valor no nulo de ACURSO.titulación simplemente se mantiene ya que en el esquema MOODLE venía así.

- b) No hace falta hacer nada, esas dependencias funcionales ya están recogidas en el esquema MOODLE.

SB12

1

Card(USUARIO, vive)= (1,1)
 Card(USUARIO, pide)= (0,N)
 Card(USUARIO, prefiere)= (0,N)
 Card(USUARIO, premiado)= (0,1)
 Card(PEDIDO, contiene)= (0,N)
 Card(LINPED, de)= (0,1)

Card(LOCALIDAD, vive)= (0,N)
 Card(PEDIDO, pide)= (0,1)
 Card(PREFERENCIA, prefiere)= (0,N)
 Card(PREFERENCIA, premiado)= (0,1)
 Card(LINPED, contiene)= (1,1)
 Card(PRODUCTO, de)= (0,N)

2

Tabla inicial: T0 (a,b,c,d,e) CP (b,c) CAlt (a,b) *

Tablas ya normalizadas hasta FNBC

T0 (a,b) CP (a,b)

CAj (a) >> T2

CAj (b) >> T1

T1 (b,d) CP (b)

CAj (d) >> T11

T11 (d,e) CP (d)

T2 (a,c) CP (a)

* también se podría haber empezado con CP (a,b) CAlt (b,c)

3

ED (d1, d2) CP (d1)

EF (d1, f1, f2)

CP (d1)

CAj (d1) >> ED

EC (d1, c1, a1, a2)

CP (d1)

CAj (d1) >> ED

CAlt (a1, a2)

CAj (a1, a2) >> EA

EA (a1, a2, d1)

CP (a1, a2)

CAj (d1) >> EC

R1 (a1, a2, d1, b1)

CP (a1, a2, d1, b1)

CAj (a1, a2) >> EA

CAj (d1, b1) >> EB

EB (d1, b1, b2)

CP (d1, b1)

CAj (d1) >> ED

4

define alias P1 para PRODUCTO

define alias P2 para PRODUCTO

P1 [prod, marca] -

(P1 x P2 donde P1.pvp < P2.pvp [P1.prod, P1.marca])

[marca]