

Ejercicio1

Apartado a

R1 (a,b,c,d,x,y,d,e); R2 (d,e,f)

R1.(d,e) not null

$$R1.(d,e) \rightarrow R2.(d,e)$$

Apartado b

R1 (a,b,c); R2 (d,e,f,x,y,a)

$$\text{R2.a} \rightarrow \text{R1.a}$$

Apartado c

R1 (a,b,c); R2 (a,e,f,x,y)

$$R2.a \rightarrow R1.a$$

Apartado d

Error de diseño

Apartado e

Error de diseño

Apartado f

R1 (a,b,c,x,y,a')
$$\text{R1.a}' \rightarrow \text{R1.a}$$

Ejercicio 2

LOG (nif,descuento)

```
CREATE TABLE LOG (
```

nif VARCHAR2(64) PRIMARY KEY,

descuento NUMBER(10) PRIMARY KEY,

FOREIGN KEY nif REFERENCE Cliente (nif),

FOREIGN KEY descuento REFERENCE Cliente (descuento)

);

CREATE OR REPLACE TRIGGER tr_log AFTER INSERT OR UPDATE ON Cliente

FOR EACH ROW

DECLARE

```
v_nif Cliente.nif%type;
```

BEGIN

```
SELECT nif INTO v_nif FROM Cliente WHERE (nif := new.nif) AND (descuento :=
new.descuento);
```

```
IF v_nif IS NULL THEN INSERT INTO LOG VALUES (:new.nif, :new.descuento);
```

END IF;

END;

```
ALTER TABLE Cliente ADD CONSTRAINT CK_CLIENTE CHECK(
    (descuento >=(SELECT minimo FROM RDescuento WHERE estado = 1))
    AND (descuento <=(SELECT maximo FROM RDescuento WHERE estado=1))
);
```

Ejercicio 3

Una relación está en FN2 si lo está en FN2 y todo atributo o no depende de la clave primaria de la relación, o depende totalmente de la clave primaria de la relación.

Para que no se cumpla la FN2, deben existir alguna de las siguientes dependencias funcionales:

$A \rightarrow C$, $A \rightarrow D$, $B \rightarrow C$, $B \rightarrow D$

Ejercicio 4

$R1 \cap R2 = R2 - ((R1 \cup R2) - R2) = R1 - (R1 - R2)$

Ejercicio 5

Apartado a

$R1(\underline{a}, b, c); R2(\underline{d}, e, f, g, a) :: R2.a \rightarrow R1.a$

$R2.a$ como clave alterna y not null

Apartado b

$R1(\underline{a}, b, c); R2(\underline{d}, e, f, g) :: R2.d \rightarrow R1.a$

Apartado c

$R1(\underline{a}, b, c); R2(\underline{h}, i, j); R3(\underline{a}, h) :: R3.a \rightarrow R1.a; R3.h \rightarrow R2.h$

$R3.h$ como clave alterna y not null

Apartado d

Error de diseño

Apartado e

$R1(\underline{d}, e, f, g, d', e') :: R1.(d', e') \rightarrow R1.(d, e)$

$R1.(d', e')$ not null

Apartado f

$R1(\underline{a}, b, c, a') :: R1.a' \rightarrow R1.a$

$R1.a'$ como clave alterna y not null

Apartado g

$R1(\underline{h}, i, j); R2(\underline{d}, e, f, g) :: R2.d \rightarrow R1.h$

Apartado h

$R1(\underline{a}, b, c, a') :: R1.a' \rightarrow R1.a$

$R1.a'$ como clave alterna

Apartado i

$R1(\underline{a}, b, c); R2(\underline{d}, e, f, g, a) :: R2.a \rightarrow R1.a$

R2.a como clave alterna y not null

Apartado j

Igual al apartado f

Apartado k

$R1(\underline{d}, e, f, g); R2(\underline{h}, i, j); R3(\underline{d}, e, h) :: R3.(d, e) \rightarrow R1.(d, e); R3.h \rightarrow R2.h$

Ejercicio 6

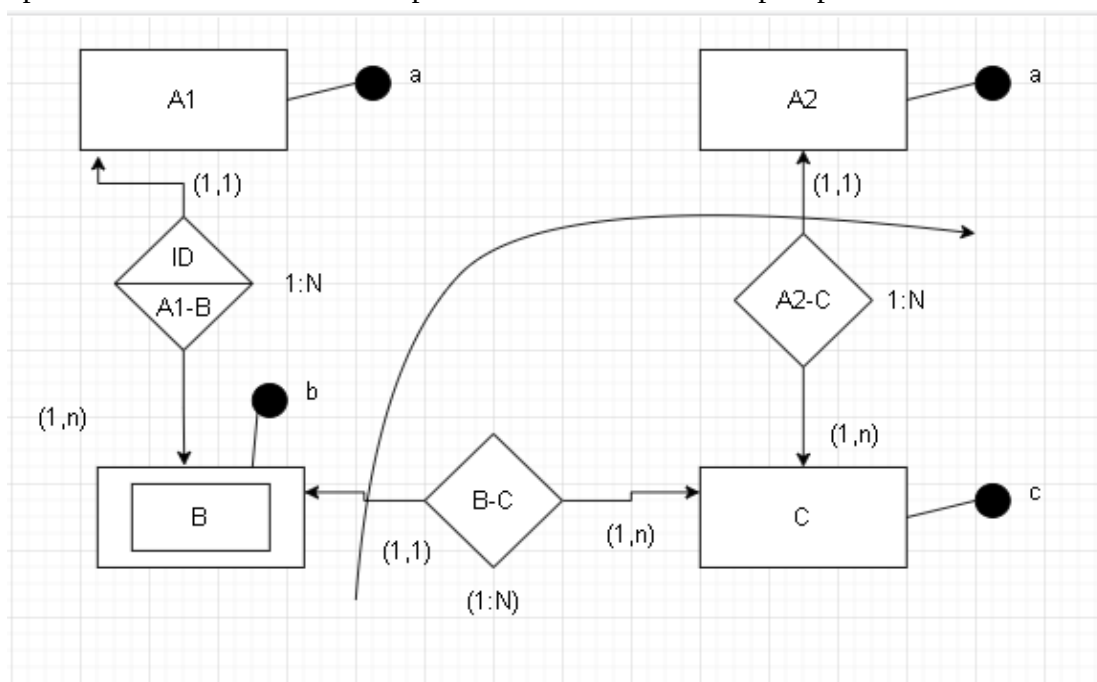
Error de diseño, pues $\{a, b, c\}$ y $\{b, c\}$ son claves traslapadas, debiendo ser $\{b, c\}$ la clave primaria

Ejercicio 7

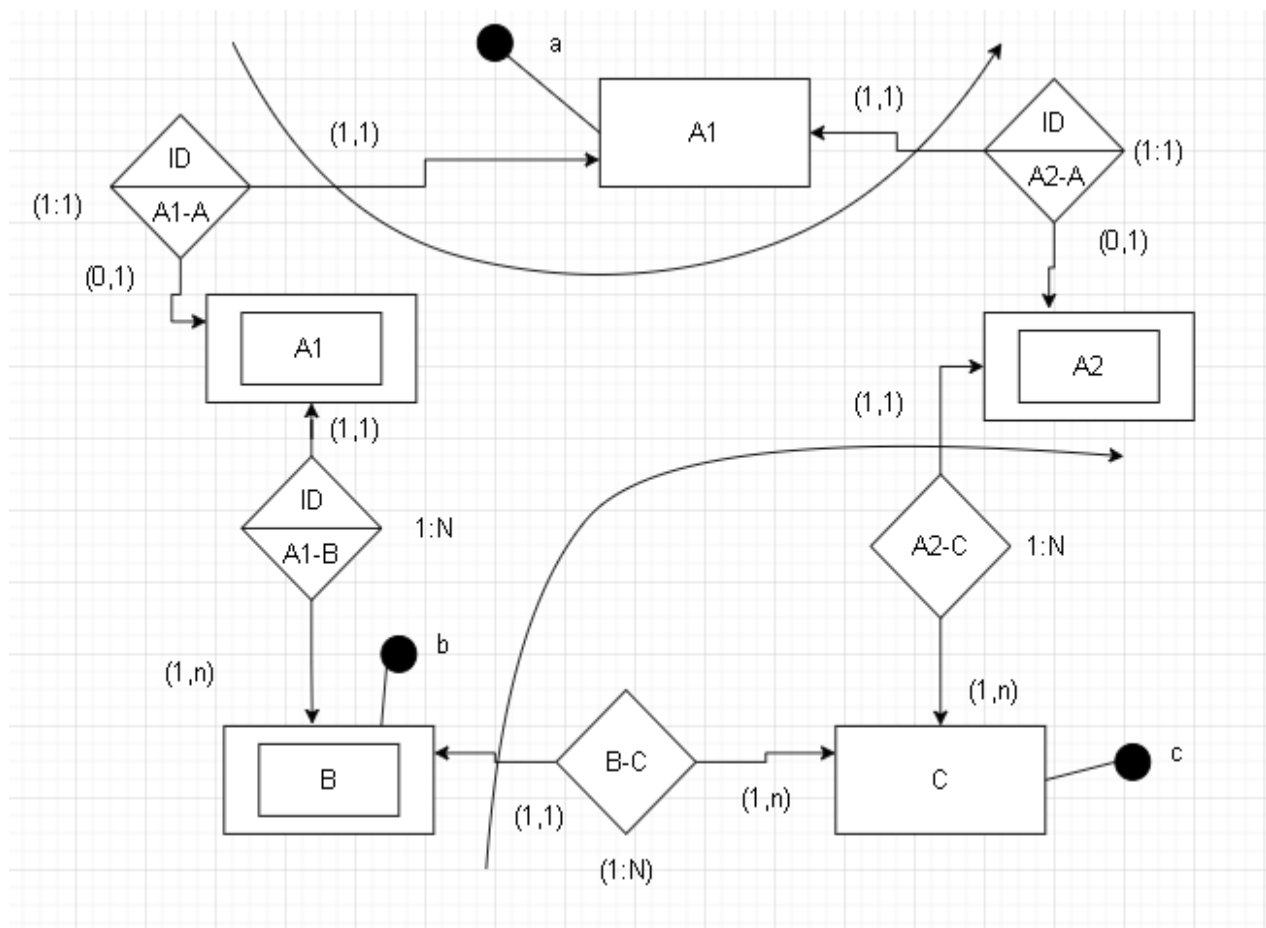
Apartado a

Para derivar el modelo relacional, podemos aplicar tanto eliminación del supertipo como eliminación del tipo de especialización.

Si aplicamos eliminación del supertipo, se produce una gran pérdida de información semántica sobre el problema. Esto se observa al aplicar la eliminación del supertipo al modelo relacional:



Si aplicamos la eliminación del tipo de especialización, se produce una menor pérdida de información semántica sobre el problema. Esto se demuestra al aplicar la eliminación del tipo de especialización al modelo relacional:



Por tanto, aplicamos la eliminación del tipo de especialización. Realizamos la transformación del modelo conceptual al modelo relacional:

$A(\underline{a}, \text{tipo}); A1(\underline{a}); A2(\underline{a}); B(\underline{b}, \underline{a}); C(\underline{c}, \underline{b}, \underline{a}, \underline{a'})$

$A1.a \rightarrow A.a; A2.a \rightarrow A.a; B.a \rightarrow A1.a; C.(b,a) \rightarrow B.(b,a); C.a' \rightarrow A2.a$

A.tipo not null

Apartado b

Si observamos el modelo relacional obtenido, observamos que se producen errores de integridad. Dichos errores de integridad son los siguientes:

- El valor del atributo A.tipo sólo puede ser 'A1' o 'A2', debido a que en el modelo relacional tenemos una especialización total exclusiva.
- Si insertamos un valor en A1, dicho valor no puede existir en A2 (y viceversa), debido al tipo de interrelación de A con A1 y A2.
- Si insertamos un valor (a,b) en C, el valor de a' debe ser NULL (y viceversa), debido al tipo de interrelación de C con B y A2.

Por tanto, debemos de aplicar una serie de procedimientos para asegurar la integridad de la base de datos:

1. Procedimiento de control del tipo de valor a insertar en A:

```
ALTER TABLE A ADD CONSTRAINT CHECK_A CHECK ((tipo = 'A1') OR (tipo = 'A2'));
```

2. Procedimiento de control de inserción o actualización de tuplas en la tabla A1:

```
CREATE OR REPLACE TRIGGER TRIGGER_A1 BEFORE INSERT OR UPDATE ON A1  
FOR EACH ROW
```

```
DECLARE
```

```
    resultado A1.a%type;
```

```
BEGIN
```

```
    SELECT a INTO resultado FROM A2 WHERE a = :new.a;
```

```
    IF resultado IS NOT NULL THEN
```

```
        RAISE_APPLICATION_ERROR (-20001, 'Error ...');
```

```
    END IF;
```

```
END;
```

3. PROCEDIMIENTO DE CONTROL DE INSERCIÓN O ACTUALIZACIÓN DE TUPLAS EN LA TABLA A2

```
CREATE OR REPLACE TRIGGER TRIGGER_A2 BEFORE INSERT OR UPDATE ON A2  
FOR EACH ROW
```

```
DECLARE
```

```
    resultado A2.a%type;
```

```
BEGIN
```

```
    SELECT a INTO resultado FROM A1 WHERE a = :new.a;
```

```
    IF resultado IS NOT NULL THEN
```

```
        RAISE_APPLICATION_ERROR (-20002, 'Error...');
```

```
    END IF;
```

```
END;
```

4. PROCEDIMIENTO DE CONTROL DE INSERCIÓN O ACTUALIZACIÓN DE TUPLAS EN LA TABLA C

CREATE OR REPLACE TRIGGER TRIGGER_C BEFORE INSERT OR UPDATE ON C
FOR EACH ROW

BEGIN

– Los atributos de a,a' y b no pueden ser o todos nulos, o todos no nulos

IF (

(:new.a IS NULL) AND (:new.b IS NULL) AND (:new.a' IS NULL)

OR ((:new.a IS NOT NULL) AND (:new.b IS NOT NULL) AND (:new.a' IS NOT
NULL)) THEN

RAISE_APPLICATION_ERROR (-20003, 'Error ...');

)

END IF;

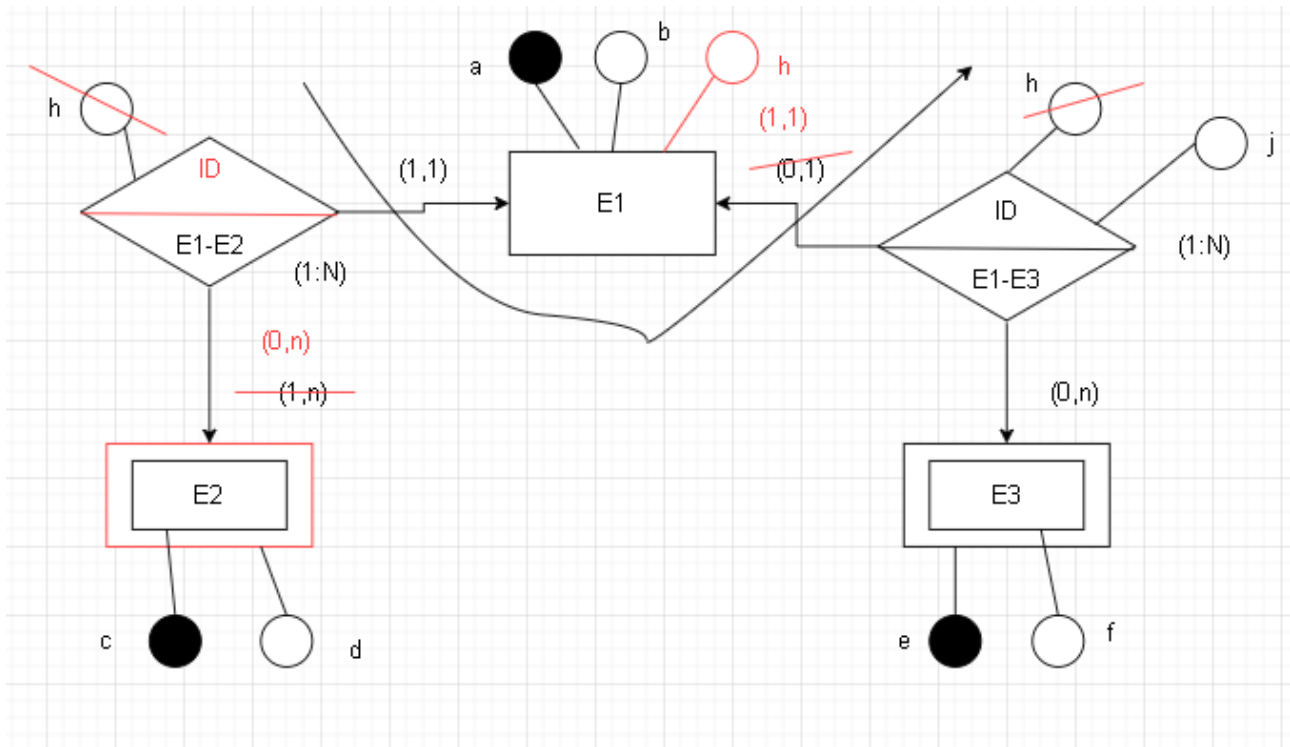
END;

Ejercicio 8

Apartado a

El diseño no es correcto, debido a que se están representando cardinalidades del tipo de interrelación entre el supertipo A con los subtipos A2 y A3 que no se corresponden a un modelo relacional con exclusividad.

Por tanto, primero debemos proponer un nuevo diseño de modelo conceptual



Apartado b

Tablas	Atributos	Claves principales	Claves alternas	Claves foraneas
E1	a,b,h	a	-	-
E2	c,d,a	c,a	-	a
E3	e,f,a	e,a	-	a

$R1(\underline{a},b,h)$; $R2(\underline{c},\underline{a},d)$; $R3(\underline{e},\underline{a},f) :: R2.a \rightarrow R1.a$; $R3.a \rightarrow R1.a$

Apartado c

Ya se encuentra en FNBC pues está en FN1 (no existen atributos múltiples en la relación) y cada determinante funcional es clave candidata de la relación.

Haría falta realizar una serie de procedimientos para asegurar la integridad de la base de datos debido a las siguientes inconsistencias:

1. Es necesario asegurar la exclusividad de E1 con E2 y E3
2. Es necesario asegurar que un valor de E2 se relacione con E3, y un valor de E3 se relacione con E3

Procedimiento de control de insercion o actualizacion de tuplas en la tabla E2

```
CREATE OR REPLACE TRIGGER TRIGGER_E2 BEFORE INSERT OR UPDATE ON E2  
FOR EACH ROW
```

```
DECLARE
```

```
    resultado E1.a%type;
```

```
    tipo E1.h%type;
```

```
BEGIN
```

```
    – El valor de h en E1 debe ser ‘E2’
```

```
    SELECT h INTO tipo FROM E1 WHERE a = :new.a;
```

```
    IF h = ‘E3’ THEN
```

```
        RAISE_APPLICATION_ERROR (-20001, ‘Error ...’);
```

```
    – El valor de a no puede estar registrado en E3
```

```
    ELSE
```

```
        SELECT a INTO resultado FROM E3 WHERE a = :new.a;
```

```
        IF a IS NOT NULL THEN
```

```
            RAISE_APPLICATION_ERROR(-20002, ‘Error ...’);
```

```
        END IF;
```

```
    END IF;
```

```
END;
```

Procedimiento de control de insercion o actualizacion de tuplas en la tabla E3

```
CREATE OR REPLACE TRIGGER TRIGGER_E2 BEFORE INSERT OR UPDATE ON E3  
FOR EACH ROW
```

```
DECLARE
```

```
    resultado E1.a%type;
```

```
    tipo E1.h%type;
```

```
BEGIN
```

```
    – El valor de h en E1 debe ser ‘E3’
```

```
    SELECT h INTO tipo FROM E1 WHERE a = :new.a;
```

```
    IF h = ‘E2’ THEN RAISE_APPLICATION_ERROR (-20003, ‘Error ...’);
```

```
    – El valor de a no puede estar registrado en E2
```

```
    ELSE
```

```
        SELECT a INTO resultado FROM E2 WHERE a = :new.a;
```

```
        IF a IS NOT NULL THEN RAISE_APPLICATION_ERROR(-20004, ‘Error ...’);
```

```
        END IF;
```

```
    END IF;
```

```
END;
```


Ejercicio 10

La relación R está en FN1 debido a que no existen atributos múltiples en la relación.

Dado que los atributos a y b pertenecen al mismo dominio, se cumple $a=c$

Dada la dependencia funcional $R.a \rightarrow R.*$, sabemos que R.a es clave candidata de la relación.

Realizamos la construcción del modelo relacional con la información aportada:

R1 (a,b,c,e)

R2 (b,d)

$R1.c \rightarrow R1.a$

$R1.b \rightarrow R2.b$