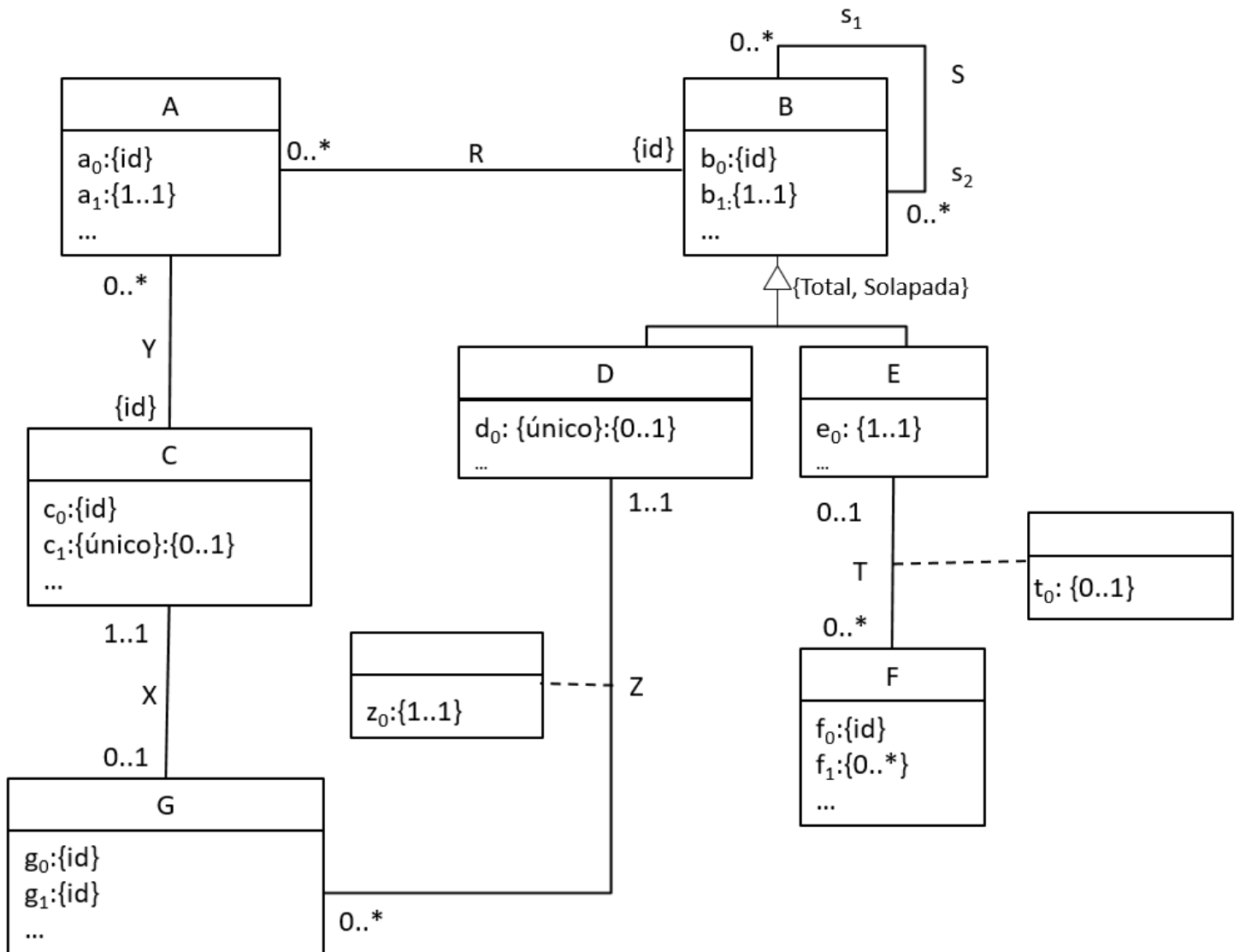


R3: UNIDADES DIDÁCTICAS 3 y 4

- 1) Realice el diseño lógico del siguiente diagrama de clases en UML para obtener un conjunto equivalente de relaciones del modelo relacional. Las restricciones que no pueda expresar en el esquema relacional, escríbalas en lenguaje natural. (1'5 puntos)



2) (0'5 puntos)

Sea el siguiente esquema de relación:

R (A: char, B: char, C:int, D: int, E: registro de {E1:int, E2:int, E3:int},
F: char, G: char, H: char, I:int)

CP: {A, B, C} VNN: {C, D, E, F, G}

Transforme la relación a un conjunto de relaciones en tercera forma normal teniendo en cuenta las siguientes dependencias:

$\{A, B\} \rightarrow \{E\}$

$\{C\} \rightarrow \{D\}$

$\{D\} \rightarrow \{F, G\}$

$\{F\} \rightarrow \{H\}$

- 3) Indica entre qué esquemas se da la independencia de datos física y la lógica. Explica muy brevemente en qué consiste la independencia lógica. (0'25 puntos)
- 4) Diseñe un diagrama de clases en UML para el sistema de información que se describe a continuación. Las restricciones que no se puedan expresar gráficamente, escríbalas en lenguaje. (1'75 puntos)

El ayuntamiento de Gotham City va a organizar la gestión de sus líneas de metro.

De cada unidad de metro (que llamaremos “tren”), identificada por su matrícula, se debe almacenar su capacidad y año de construcción.

Cada conductor de tren se identifica por su DNI y tiene asociado obligatoriamente un nombre, una dirección y, opcionalmente, una lista de números de teléfonos. Un conductor debe pasar un examen para poder conducir un tren por una línea. Cuando un conductor pasa un examen conduciendo un tren por una línea concreta de metro, queda habilitado para conducir ese tren en la línea en la que ha realizado el examen. Esta habilitación para conducir un tren por una línea tiene una fecha de caducidad, tras la cual el conductor deberá volver a examinarse.

Cada línea de metro está identificada por un número, tiene una longitud, opcionalmente una frecuencia media de paso y está compuesta por varias estaciones. Las estaciones que forman parte de una línea están numeradas consecutivamente desde la estación origen hasta la estación final de la línea. Sabemos que una misma estación puede formar parte de varias líneas. Una estación de metro se identifica por un nombre pero también necesitamos conocer su dirección. Cada estación está incluida en una zona de tarifas. Cada zona, identificada por un número, tiene asignado un color.

Cada cliente que se registra en el sistema está identificado por su DNI y tiene asociado obligatoriamente un nombre, una dirección, un número de tarjeta de crédito y, opcionalmente, una lista de números de teléfonos. Los clientes pueden comprar tarjetas de viaje. Cada tarjeta se identifica por un número único y se debe guardar el tipo de chip usado por la tarjeta. Un cliente puede recargar cualquier tarjeta (aunque no sea suya). Las recargas se identifican con el cliente que la ha realizado y un número. De cada recarga se guarda la fecha, el importe pagado, el número de viajes recargados y la zona o zonas para las que es válida la recarga.

EXAMEN DE BASES DE DATOS Y SISTEMAS DE INFORMACIÓN (21/01/20)

Soluciones:

1.

| | |
|--|---|
| A (a0, a1, ... b0, c0) CP:{a0, b0, c0} CAj:{b0} → B CAj:{c0} → C VNN:{a1} | B (b0, b1, ...) CP:{b0} VNN{b1} |
| C (c0, c1, ...) CP{c0} Uni:{c1} | D (b0, d0, ...) CP:{b0} CAj:{b0} → B Uni:{d0} E (b0, e0, ...) CP:{b0} CAj:{b0} → B VNN{e0} RI_{Total} : Los valores que aparezcan en el atributo b0 de B debe aparecer en el atributo b0 de D o en el b0 de E. |
| F (f0, ...) CP{f0} F1 (f1, f0) CP:{f1, f0} CAj:{f0} → F | G (g0, g1, ..., b0, z0, c0) CP{g0, g1} CAj:{b0} → D VNN:{b0} VNN:{z0} CAj:{c0} → C VNN{c0} Uni:{c0} |
| T (b0, f0, t0) CP{f0} CAj:{b0} → E CAj:{f0} → F VNN:{b0} | S (b0_s1, b0_s2) CP:{b0_s1, b0_s2} CAj:{b0_s1} → B(b0) CAj:{b0_s2} → B(b0) |

2.

| | |
|--|--|
| R(A:char, B:char, C:int, I:int) CP:{A,B,C} CAj:{A,B} → R1 CAj:{C} → R2 | R1(A:char, B:char, E1, E2, E3) CP:{A,B} VNN:{E1, E2, E3} RI: Toda tupla de R1 debe aparecer en R |
| R2(C:int, D:int) CP:{C} VNN:{D} CAj:{D} → R3 RI: Toda tupla de R2 debe aparecer en R | R3(D:int, F:char, G:char) CP:{D} VNN:{F,G} CAj:{F} → R4 RI: Toda tupla de R3 debe aparecer en R2 |
| R4(F:char, H:char) CP:{F} VNN:{H} RI: Toda tupla de R4 debe aparecer en R3 | |

3.

Independencia física: entre el esquema lógico y el esquema físico.

Independencia lógica: entre los esquemas externos y el esquema lógico

Los esquemas externos y los programas de aplicación no deben verse afectados por modificaciones del esquema lógico sobre datos que no usan.

4.

