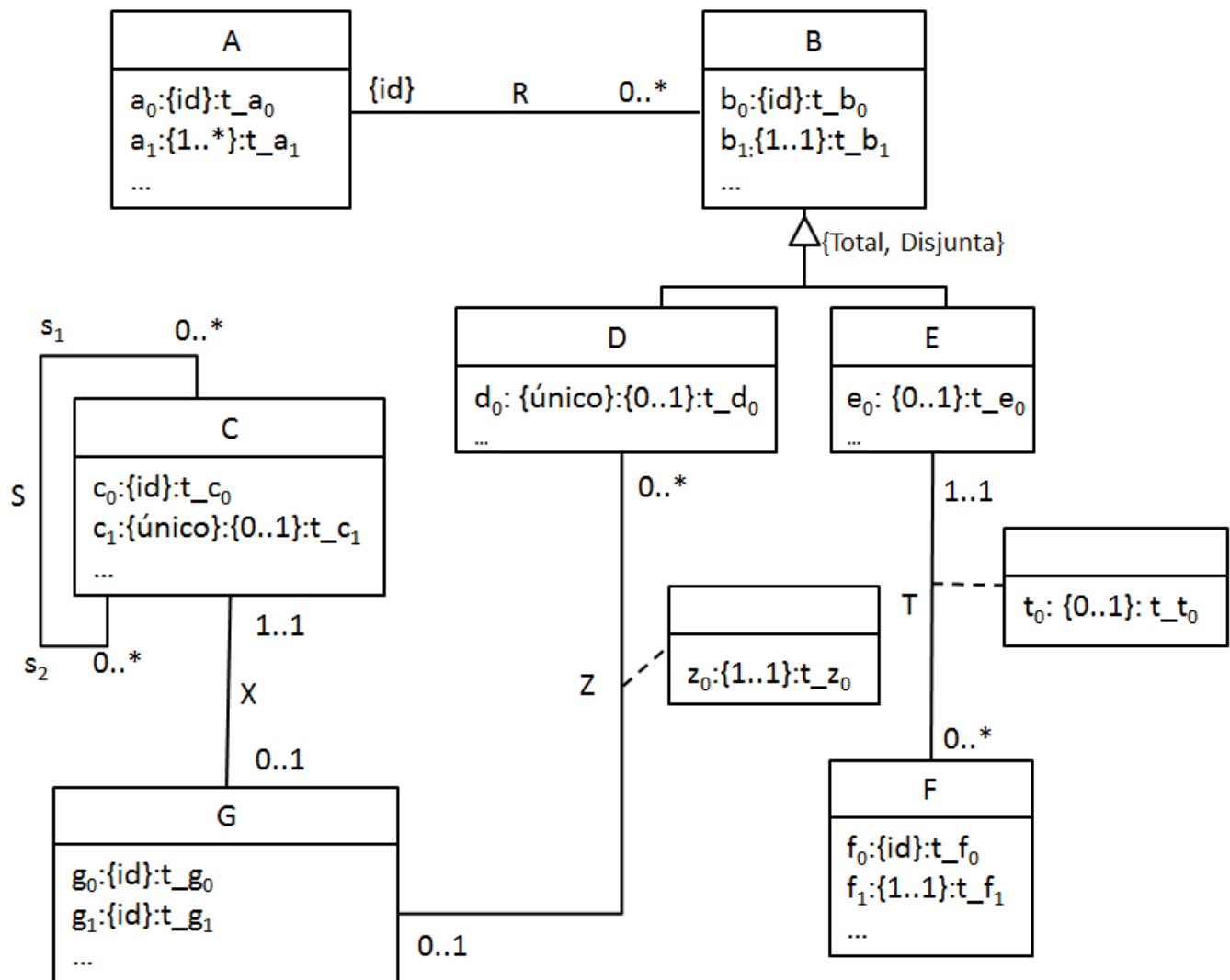


## PARCIAL 2: UD4 Y UD5

- 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 puedan expresarse en el esquema relacional, escríbalas en lenguaje natural.

(1'5 puntos)



2) 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, escribálas en lenguaje natural.

(1'5 puntos)

Se desea diseñar una base de datos para la gestión de las reservas de una empresa de alquiler de vehículos.

La empresa dispondrá de sucursales repartidas por todo el país, almacenando un código de sucursal, que la identifica, y una dirección que no se puede repetir. En las sucursales habrá diferentes vehículos que podrán ser alquilados. De cada vehículo se desea conocer su matrícula, que identifica al vehículo, la marca y el modelo del mismo. Se dispone de dos tipos de vehículos: coches (donde interesa saber el número de plazas) y furgonetas (donde interesa saber el volumen de carga). Los vehículos estarán clasificados según su categoría. Cada categoría tendrá un nombre identificador y un precio de alquiler diario.

En las sucursales también hay empleados, de los que se conoce su dni, que los identifica, su nombre, apellidos, dirección, población, código postal y cargo. Todos estos datos son obligatorios.

Los clientes pueden realizar reservas. De un cliente se conoce su dni, que le identifica, su nombre, apellidos, dirección, población, código postal, fecha de obtención del carnet de conducir y el número de una tarjeta de crédito. Todos estos datos son obligatorios.

Cada reserva que se realiza se codifica con un valor único. También se conoce el cliente que la realiza, la categoría de vehículo que desea, la fecha y sucursal de recogida y la fecha y sucursal de devolución. Todos estos datos son obligatorios. Un cliente no puede realizar dos reservas con la misma fecha de recogida.

Cuando se produce la entrega del vehículo a partir de una reserva, se anota el empleado que ha realizado la entrega, qué vehículo se ha asignado, el kilometraje y la cantidad de combustible que tiene. Estos datos son obligatorios para cualquier entrega. Cada entrega se corresponde con una única reserva.

Por último, cuando se devuelve el vehículo, se registra la devolución, se anota el empleado encargado de su recepción, el kilometraje, la cantidad de combustible que queda y una lista de desperfectos que haya podido sufrir el vehículo. Cada devolución se corresponde con una única entrega.

3) Sea el siguiente esquema de relación:

R(A: char, B: char, C: int, D: char, E: int, F: char, G: int, H: char)

CP: {A, B, C}

VNN: {D, E, F, G, H}

A partir de las dependencias de clave primaria más las que aparecen a continuación, transforme la relación a un conjunto de relaciones en tercera forma normal

$\{A\} \rightarrow \{D\}$

$\{E\} \rightarrow \{F\}$

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

$\{G\} \rightarrow \{H\}$

(0'5 puntos)

4) Dado el siguiente esquema de una base de datos donde

```
CREATE TABLE Curso (  
    codigo CHAR(5) CONSTRAINT pk_curso PRIMARY KEY DEFERRABLE,  
    nombre VARCHAR(50) NOT NULL,  
    descripcion VARCHAR(50) NOT NULL,  
    edad_minima INTEGER NOT NULL);
```

```
CREATE TABLE Alumno (  
    exp char(5) CONSTRAINT pk_alumno PRIMARY KEY DEFERRABLE,  
    nombre VARCHAR(50) NOT NULL,  
    edad INTEGER NOT NULL,  
    curso CHAR(10) NOT NULL  
        CONSTRAINT fk_alumno_curso REFERENCES Curso(codigo)  
        DEFERRABLE);
```

Donde:

- La relación **Curso** almacena los datos de un curso: código, nombre, descripción y la edad mínima que deben tener los alumnos para poder inscribirse en el curso.
- La relación **Alumno** almacena la información de la inscripción de un alumno en un curso: número de expediente (exp), nombre del alumno, edad y código del curso en el que se quiere inscribir (curso).

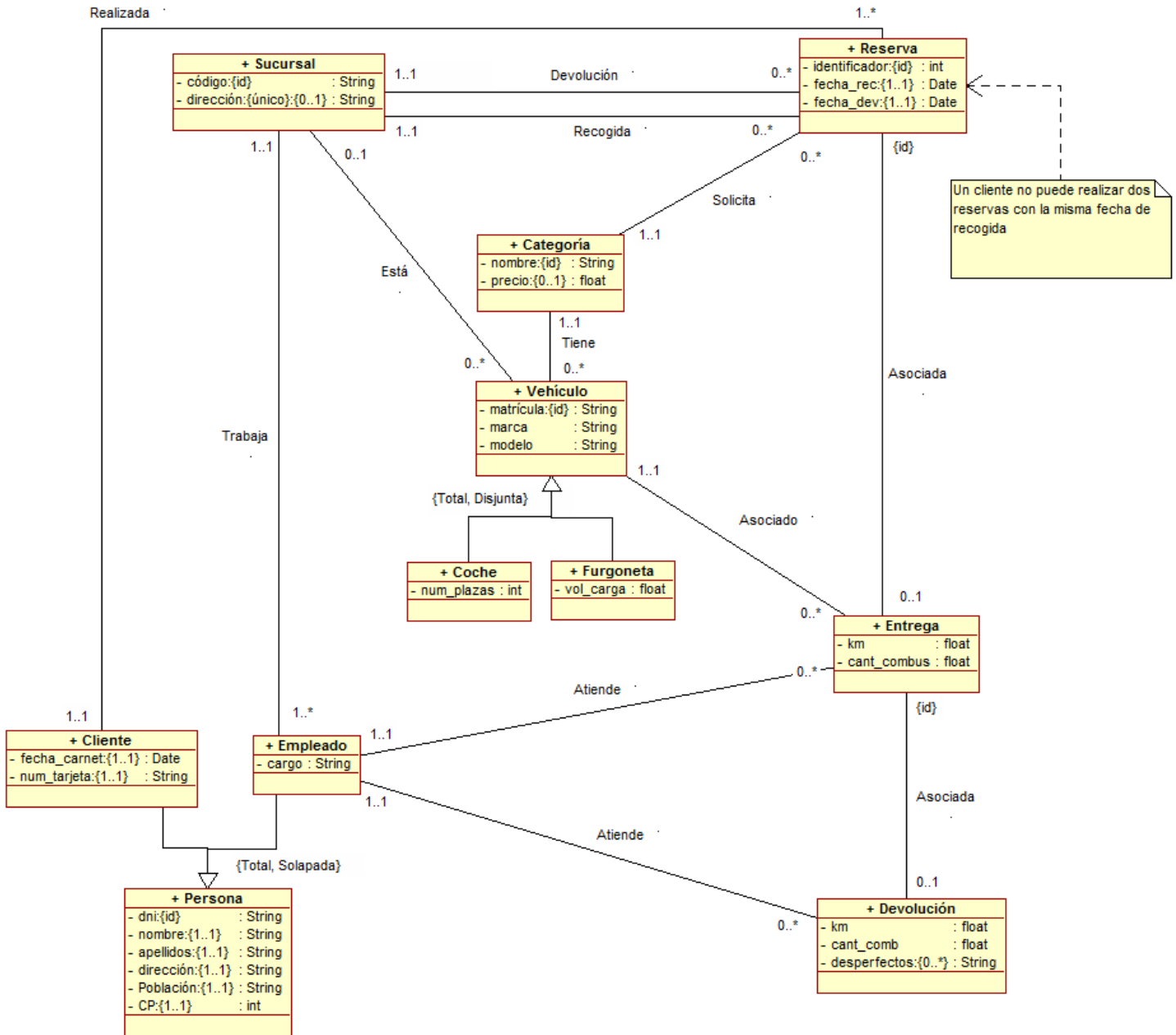
Implementar un disparador (trigger) en PL-SQL de Oracle que se active cuando se produce una inserción en la relación Alumno y compruebe que la edad del alumno es la adecuada para el curso en el que se quiere inscribir. En caso de que la edad sea inferior a la edad mínima del curso, el disparador debe impedir la inserción.

(0'5 puntos)

1.-

<p><b>A</b>(a0:t_a0, ...) CP:{a0}</p> <p><b>A1</b>(a0:t_a0, a1:t_a1) CP:{a0, a1} CAj:{a0} → A</p> <p><b>RI<sub>MinA1</sub></b>: Todo valor que aparezcan en el atributo a0 de A debe aparecer en el atributo a0 de A1.</p> <p><b>B</b>(a0:t_a0, b0:t_b0, b1:t_b1,...,) CP:{a0, b0} CAj:{a0} → A VNN:{b1}</p> <p><b>D</b>(a0:t_a0, b0:t_b0, d0:t_d0, ...) CP:{a0, b0} CAj:{a0, b0} → B Uni:{d0}</p> <p><b>E</b>(a0:t_a0, b0:t_b0, e0:t_e0, ...) CP:{a0, b0} CAj:{a0, b0} → B</p> <p><b>RI<sub>Disjunta</sub></b>: Ningún par de valores que aparezca en los atributos {a0, b0} de D puede aparecer en valores del par de atributos {a0, b0} de E.</p> <p><b>RI<sub>Total</sub></b>: Todo par de valores que aparezca en los atributos {a0, b0} de B debe aparecer en los atributos {a0, b0} de D o de E.</p>	<p><b>F</b>(f0:t_f0, f1:t_f1, ..., a0:t_a0, b0:t_b0, t0:t_t0) CP{f0} VNN:{f1} CAj:{a0, b0} → E VNN:{a0, b0}</p> <p><b>G</b>(g0:t_g0, g1:t_g1, ..., c0:t_c0) CP{g0, g1} Uni:{c0} VNN:{c0} CAj:{c0} → C</p> <p><b>Z</b>(a0:t_a0, b0:t_b0, g0:t_g0, g1:t_g1, z0:t_z0) CP{a0, b0} VNN:{z0} CAj:{g0, g1} → G CAj:{a0, b0} → D VNN:{g0, g1}</p> <p><b>C</b>(c0:t_c0, c1:t_c1, ...) CP{c0} Uni:{c1}</p> <p><b>S</b>(c0_s1:t_c0, c0_s2:t_c0) CP:{c0_s1, c0_s2} CAj:{c0_s1} → C(c0) CAj:{c0_s2} → C(c0)</p>
---	--

2.-



3.-

<p>R(A: char, B: char, C: int, G: int)</p> <p>CP: {A, B, C}</p> <p>VNN: {G}</p> <p>CAj:{A} → R1</p> <p>CAj:{B, C} → R2</p> <p>CAj:{G} → R4</p> <p>R1(A: char, D: char)</p> <p>CP: {A}</p> <p>VNN: {D}</p>	<p>R2(B: char, C: int, E: int)</p> <p>CP: {B, C}</p> <p>VNN: {E}</p> <p>CAj:{E} → R3</p> <p>R3(E: int, F: char)</p> <p>CP: {E}</p> <p>VNN: {F}</p> <p>R4(G: int, H: char)</p> <p>CP: {G}</p> <p>VNN: {H}</p> <p>Todo valor de A en R1 deberá existir en R</p> <p>Todo valor B,C en R2 deberá existir en R</p> <p>Todo valor E en R3 deberá existir en R2</p> <p>Todo valor G en R4 deberá existir en R</p>
---	--

4.-

```
CREATE OR REPLACE TRIGGER T_ins_alumno
BEFORE INSERT ON Alumno
FOR EACH ROW
DECLARE
    minimo INTEGER;
BEGIN
    SELECT edad_minima INTO minimo
    FROM Curso WHERE codigo = :new.curso;
    IF minimo > :new.edad THEN
        RAISE_APPLICATION_ERROR(-20000, 'La edad del alumno
        no puede ser inferior a la edad mínimo del curso');
    END IF;
END;
```