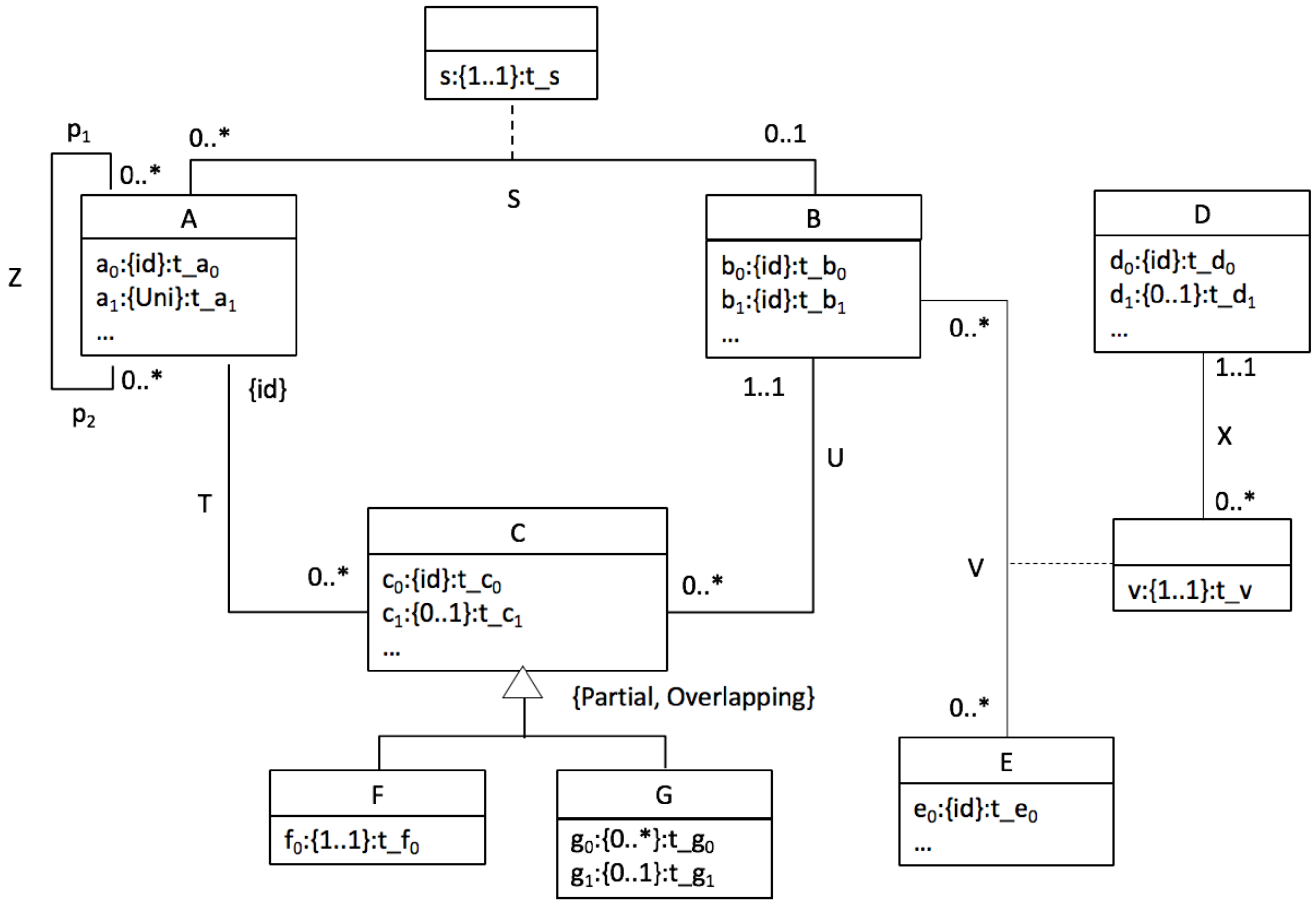
“BASES DE DATOS Y SISTEMAS DE INFORMACIÓN” EXAM 10/01/2019

UNIT 4

1. Perform the logical design of the following UML class diagram in order to obtain the equivalent set of relations in the relational model. Those constraints that cannot be expressed graphically can be written in natural language (English or Spanish). **(1.75 points).**



2. Consider the following relational schema:

R (**A**: *int*, **B**: *char*, **C**: *char*, **D**: *set of integers*, **E**: *int*, **F**: *text*, **G**: *int*, **H**: *int*)
PK: {A, B, C}
NNV: {D, E, F, G, H}

From the dependencies shown below, transform the relation to a set of relations in third normal form. **(0.5 points).**

{B} → {G} {A, B} → {F} {G} → {H}

3. Design a UML class diagram for the following information system. Those constraints that cannot be expressed graphically, can be written in natural language (English or Spanish). **(1.75 points)**

The North Pole Company and its CEO, S. Claus (S.C.), want to organize their information in a new database. S.C. wants to save the information of the elves working at the company. Each elf is uniquely identified by a number, and has a name (that is unique in the company) and a role: toys manufacturer elf or delivery-elf. Every delivery-elf is assigned to a city and he/she will be distributing toys only to that city.

In order to organize the distribution of toys, S.C. needs to know, for each country in which the company has to distribute toys, an identifier code, the name of the country, and the population density. For each country, he also wants to know the cities that his team must visit. A city has an identifier that is unique within its country, a name, and the number of children living in the city. Every city will be assigned to one or more delivery-elves. To synchronize the delivery process, when a city is assigned to a delivery-elf, S.C. will also have to assign the time at which the delivery-elf must be in that city on the night of the 24th of December.

The company manufactures several models of toys. Each model of toy has an identifier code, a name, and a recommended age. For each model of toy S.C. also need to know the number of units that must be manufactured to meet the demand of the children. Each unit will be identified by a serial number that will be correlated in each model of toy. S.C. would like to store for each unit, if possible, which elf has manufactured the unit. When a manufacturer elf finishes a unit, S.C. has to save into the database the date and time on which the unit has been finished.

Finally, the company needs to identify the children who have placed orders to the company. The company will save the identifier of each child, his/her name, the city where she/he lives in, his/her address, the toys he/she has requested and if the child had behaved well or badly during the last year. Only good children will be assigned one or more units of the toys she/he has requested. It is necessary to know which unit has been assigned to each child. Only units that have been completely finished can be assigned. A child could not receive two units of the same model of toy.

SOLUTIONS

1.-

A (a_0:t_a$_0$, a_1:t_a$_1$,...)

PK:{a_0}

Uni:{a_1}

B (b_0:t_b$_0$, b_1:t_b$_1$,...)

PK:{b_0, b_1}

D (d_0:t_d$_0$, d_1:t_d$_1$,...)

PK:{d_0}

E (e_0:t_e$_0$,...)

PK:{e_0}

F (c_0: t_c$_0$, a_0: t_a$_0$, f_0: t_f$_0$)

PK:{c_0, a_0}

FK:{c_0, a_0} $\rightarrow$ C(c_0, a_0)

NNV:{f_0}

G (c_0: t_c$_0$, a_0: t_a$_0$, g_1: t_g$_1$)

PK:{c_0, a_0}

FK:{c_0, a_0} $\rightarrow$ C(c_0, a_0)

G0 (c_0: t_c$_0$, a_0: t_a$_0$, g_0: t_g$_0$)

PK:{c_0, a_0 , g_0}

FK:{c_0, a_0} $\rightarrow$ G(c_0, a_0)

Z (a_0_p1:t_a$_0$, a_0_p2:t_a$_0$)

PK:{a_0_p1, a_0_p2}

FK:{a_0_p1} $\rightarrow$ A(a_0)

FK:{a_0 _p2} $\rightarrow$ A(a_0)

C (c_0: t_c$_0$, a_0: t_a$_0$, c_1: t_c$_1$, b_0:t_b$_0$ b_1:t_b$_1$,...)

PK:{c_0, a_0}

NNV:{b_0, b_1}

FK:{a_0} $\rightarrow$ A

FK:{b_0,b_1} $\rightarrow$ B(b_0, b_1)

S (a_0: t_a$_0$, b_0: t_b$_0$, b_1: t_b$_1$, s: t_s)

PK:{a_0}

NNV:{b_0, b_1, s}

FK:{a_0} $\rightarrow$ A

FK:{b_0 ,b_1} $\rightarrow$ B(b_0, b_1)

V (b_0: t_b$_0$, b_1: t_b$_1$, e_0: t_e$_0$, v: t_v, d_0:t_d$_0$)

PK:{b_0, b_1, e_0}

NNV(d_0, v)

FK:{b_0,b_1} $\rightarrow$ B(B_0, b_1)

FK:{e_0} $\rightarrow$ E

FK:{d_0} $\rightarrow$ D

2.-

R (A: int, B: char, C: char, E: int)

PK: {A, B, C}

FK: {A, B} → R3

FK: {B} → R2

NNV: {E}

R1 (A: int, B: char, C: char, D: int)

PK: {A, B, C, D}

FK: {A, B, C} → R

IC: Every (A, B, C) in R1 must appear in R

R2 (B: char, G: int)

PK: {B}

FK: {G} → R22

NNV: {G}

R3 (A: int, B: char, F: text)

PK: {A,B}

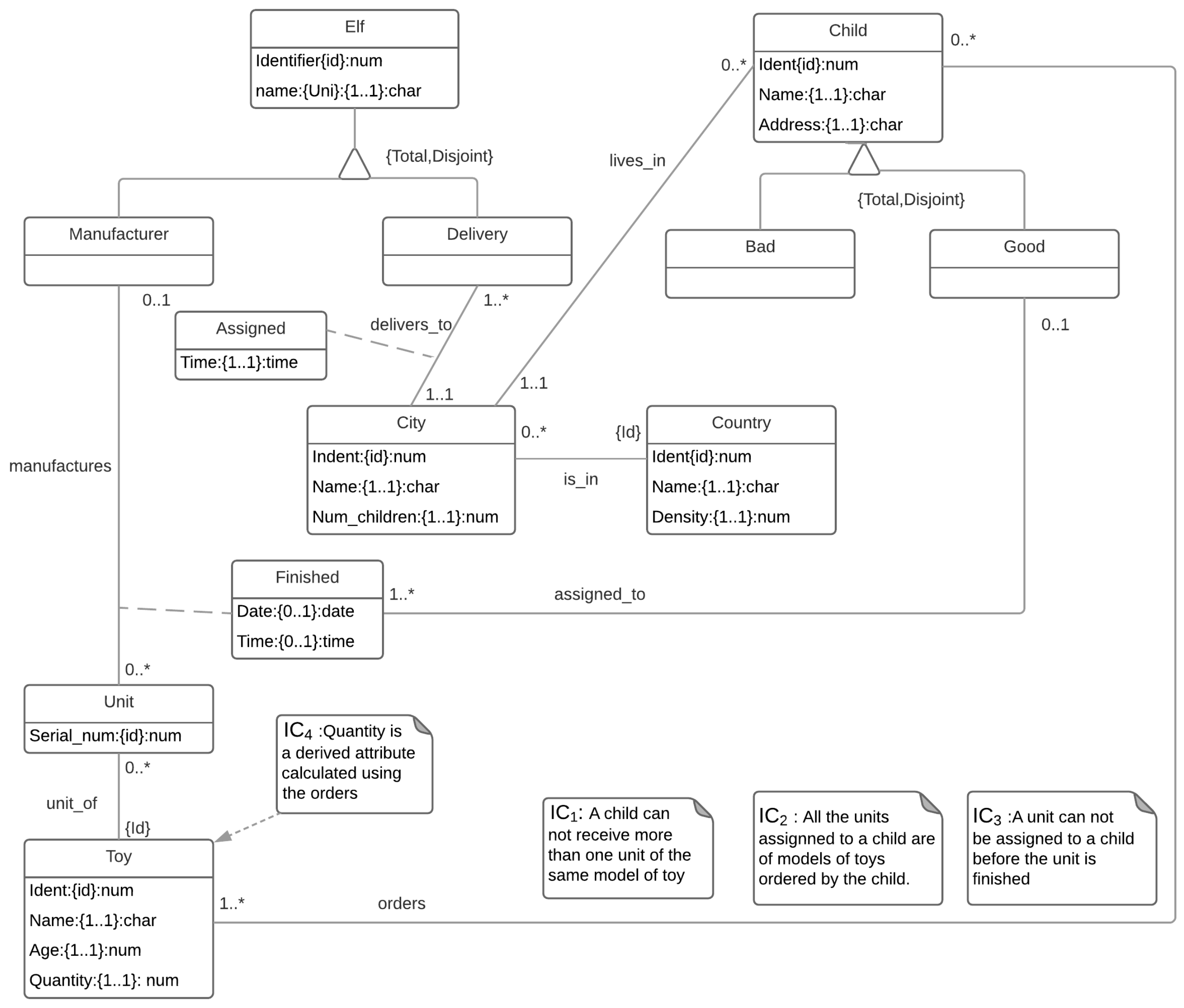
NNV: {F}

R22 (G: int, H: int)

PK: {G}

NNV: {H}

3.-


Elf
Identifier{id}:num
name:{Uni}:{1..1}:char
{Total,Disjoint}
Manufacturer
Delivery
Child
Ident{id}:num
Name:{1..1}:char
Address:{1..1}:char
0..*
0..*
lives_in
{Total,Disjoint}
Bad
Good
0..1
1..*
Assigned
Time:{1..1}:time
delivers_to
1..1
1..1
0..1
City
Indent:{id}:num
Name:{1..1}:char
Num_children:{1..1}:num
0..*
{Id}
is_in
Country
Ident{id}:num
Name:{1..1}:char
Density:{1..1}:num
manufactures
Finished
Date:{0..1}:date
Time:{0..1}:time
1..*
assigned_to
0..*
Unit
Serial_num:{id}:num
0..*
unit_of
{Id}
Toy
Ident:{id}:num
Name:{1..1}:char
Age:{1..1}:num
Quantity:{1..1}: num
1..*
orders
IC$_4$:Quantity is a derived attribute calculated using the orders
IC$_1$: A child can not receive more than one unit of the same model of toy
IC$_2$: All the units assignned to a child are of models of toys ordered by the child.
IC$_3$:A unit can not be assigned to a child before the unit is finished