Réponse de l’exercice 1 :

Clearly parameters are discount type (DT), number of tickers (N); nothing is said about the day of the week (DW) and the hour (HD) thus we assume tickets can only be sold at the entry the day the center is open and plus the software automatically access the hour of the day. Finally we assume it is the employee that will verify the age and thus child or elderly obtain 100% discount. We have to model the set of possible “legal” discounts; this (as well as the hour and day) is a kind of internal/environmental parameter. Normal discounts (aka those just due to the type of entry) are 10%, 20% and 100%. But we must also consider families. Families are assumed to be 2 or more persons and thus possible discounts are 20%, 30%, 40% and 50% when we have 5 individuals or plus. This is because family discount has an upper limit: 50%. Last, we must not forget that children under 10, and elderly > 65, have 100% discount!

We also must be careful to check that no ticket is sold for Monday and Friday and no discount is given over the weekend or between 12:00 and 15:00. Overall we end up with a table of possible discounts as:

DS0	**0**
DS1	**20**
DS2	**30**
DS3	**40**
DS4	**50**
DS5	**100**

The only reaming problem is that not all combinations are feasible! For the parameter DT we must just model the type and thus the only interesting characteristic is if it is legal or not. For that characteristic we can choose the following choices:

DT1 family [prop fam]
DT2 student [prop student]
DT3 elderly [prop eld]
DT4 regular [prop no_disc]
DT5 child [prop child]
DT6 group [prop group]
DT7 other strings [error]

Notice we added explicitly the option child and elderly to ensure we do not miss the 100% discount

For the number of tickets we should model the legal versus non legal quantities thus the only interesting category is in range or not; matter of fact we must have as legal choices 1,2,3,4, … 9, 10-20 thus we can say:

N0 0 [error]
N1 1 [prop single]
N2 2 [prop n2]
N3 3 [prop n3]
N4 4 [prop n4]
N5 5 [prop n5]
N6 6-9 [prop many]
N7 10-20 [prop group] [if !fam]
N8 >20 [error]

In a way we assume no more than 9 individuals make a family.

For the day we must model that is today as we assume tickets are sold on the spot not for the future. Clearly we have two days that generate an error second characteristic the day of the week

DWW0 Monday or Friday [error]
DWW1 Tuesday to Wednesday [prop disc, ok]
DWW2 Saturday and Sunday [prop no_disc, weekend,ok]
DWW3 Thursday [prop free, ok, disc]

We must take care of the hours assuming it opens at 9 and closes at 21:

HD0 9-12 [prop no_disc]
HD1 12-21 [prop disc] [if !weekend]

Notice we assume even a no one has no discount over the weekend but the Thursday is forced to be free. This may be debatable however the text is clear Thursday free access.

The constraints plays a big role with the discount computation:

D0 D=0 [if (no_disc || many &&!fam) && ok]
D1 D=20 [if (group || fam && n2 || student) && disc && ok]
D2 D=30 [if (fam&& n3) && disc && ok]
D3 D=40 [if (fam&& n4) && disc && ok]
D4 D=50 [if (fam && (n5||many)) && disc && ok]
D5 D=100 [if (child || elderly || free) && disc && ok]
D6 D>100 or 50 <D<100 [error]

There are different ways to define constraints here we overdone, the ok is not really needed.

Or the fist question we note that the number of tickets is the most numerous class thus we will need to write 8 frames, these are make like N3 DT1 DWW2 HD0 D3. The second question asks for the base choice and we need to define the base we can assume a single is the base thus N1 DT4 DWW1 HD0 D0 then keep fix all but one category and we make choice of the category vary.

Overall, although, in theory, we will have 7 x 9 x4 x 2 x 8 but many of these are actually leading to error and thus not expanded.