

Exercice 1 :

1. Annuaire(nom *String*, prenom *String*, telephone *String*)
Il peut être plus judicieux de définir le numéro de téléphone en *String* car cela permettra de prendre en compte ceux qui commencent par un +.
2. (a) Oui, l'ensemble vide est une relation valide.
(b) Oui.
(c) Non, car les deux numéros de téléphone sont identiques.
(d) Oui, car les noms et prénoms ne forment pas une clé primaire.
(e) Non, car la seconde entité est mal formée.
(f) Non, car le téléphone est un entier et non une chaîne de caractère.

Exercice 2 :

1. La clé primaire *nom-prenom* ne permet pas de garantir l'unicité. On peut utiliser un identifiant supplémentaire (numéro d'étudiant). La remarque est valable également dans la relation *Notes*.
 - Eleves(nom *String*, prenom *String*, num_etudiant *String*)
 - Matieres(intitule *String*, id_matiere *Integer*)
 - Notes(num_etudiant *String*, id_matiere *Integer*, notes *Integer*)Les attributs *num_etudiant* et *id_matiere* sont des clés étrangères qui référencent respectivement le numéro étudiant de l'élève et l'identifiant de la matière. Il ne peut y avoir qu'une note par matière existante et par élève existant.
2. — {("Dupont", "Jean", "A2309"), ("Durand", "Jacques", "ER450")}
— {("NSI", 1), ("EPS", 2)}
— {("A2309", 1, 18), ("A2309", 2, 15)}

Exercice 3 :

1. — Stations(nom *String*, id *Integer*)
— Lignes(numero *Integer*, couleur *String*)
La réflexion initiale du schéma est importante. A priori, définir le domaine du *numéro* de la ligne peu paraître correct. Cependant si une ligne annexe se construit et que l'on nommerait 13-A et 13-B, le modèle ne tient plus.
2. Horaires(numero *Integer*, id_station *Integer*, heure *Integer*)
Nous pourrions également décider de définir un *id* pour clé primaire.