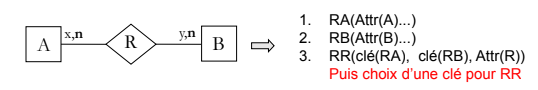
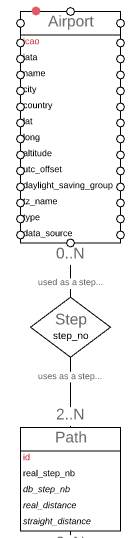
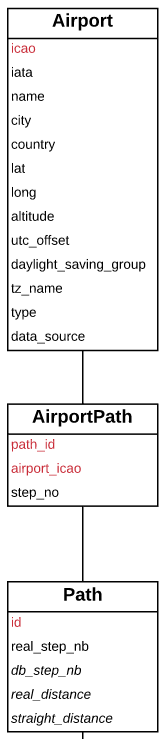
**Passage E/A à diagramme relationnel :**

**CAS d’associations binaires N/M**



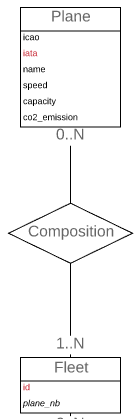
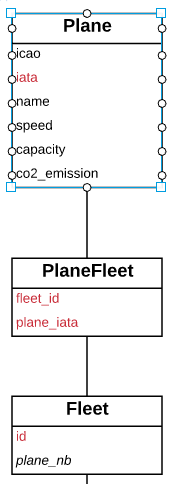


Ici nous allons créer une relation Airport, contenant tous les attributs de l’entité Airport, sa clef primaire sera donc l’ID de Aiport, icao.

De même, nous allons créer une relation Path, contenant tous les attributs de l’entité Path, sa clef primaire sera donc l’ID de Path, id

Et nous allons créer une 3ème relation issue de l’association de ces deux entités, elle aura pour but de lier de façon ordonné les aéroports avec les chemins dont ils font partie. La relation que nous appellerons Step, sera composé des clefs primaires de Airport et Path ainsi qu’un attribut propre à l’association Step, step\_no, donnant la place de l’avion dans le chemin (ex : pour un chemin x, l’aéroport y a le step\_no : 2, il est la première escale du chemin.

Ainsi on obtient les relations suivantes :

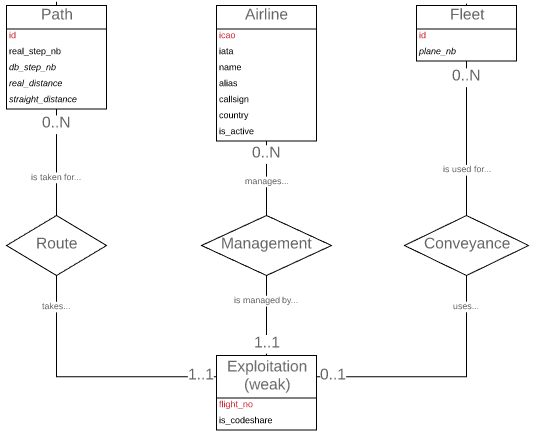
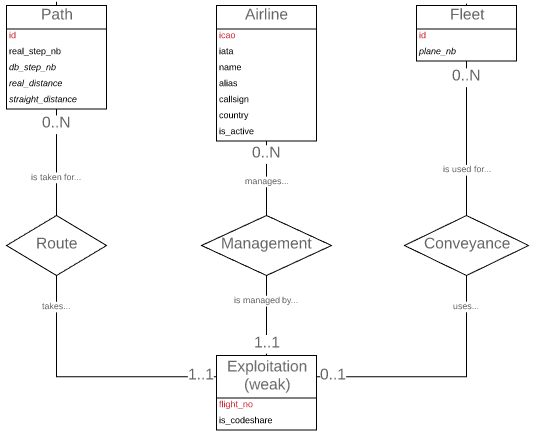


Nous allons appliquer la même méthode pour cette autre relation binaire N/M.

On peut toutefois remarquer une différence, ici, l’association Composition n’a pas d’attribut, effectivement, l’ordre des avions dans la flotte n’a pas d’importance. Ainsi la relation PlaneFleet a pour but lier des avions à leurs flottes.

**Cas d’associations binaires 1/N**





Nous allons créer une relation issue l’entité Path, contenant tous ses attributs.

.

**Contrainte d’intégrité :**

Real\_step\_nb >= db\_step\_nb

Real distance >= straight distance