

Notes de Clarification

Remarques générales

- Tous les attributs sont NOT NULL par défaut.
- Tous les héritages sont exclusifs.

Clarifications concernant les vols

- Un vol est géré par une seule compagnie.
- Une escale compte comme deux vols distincts, nous comptons donc juste le vol associé à notre aéroport.
- Il n'y a pas d'identifiant unique pour le passager, alors que c'est le cas pour les membres du personnel.
- Il est possible d'avoir un vol à vide (vol de marchandise), ce qui implique donc qu'il n'y a pas de passagers.
- La provenance et la destination sont représentées par les aéroports (Elles seront donc représentées sous une forme String et non une classe).
- L'heure de départ et d'arrivée des vols sont des datetime permettant d'obtenir l'année.
- Les identifiants de deux vols de même destination se passant à la même date mais à des années différentes sont différents. Un vol a donc un identifiant qui lui est unique (il sera donc la clé primaire).
- L'heure de départ est supérieure à l'heure d'embarquement.
- Il faut vérifier pour un vol si son avion (son modèle) peut effectivement embarquer/débarquer sur une porte donnée.

Clarifications concernant les compagnies

- Un Modèle a forcément une compagnie de fabrication.
- Le nom de la compagnie est unique (il sera donc la clé primaire).

Autres Clarifications

- Un passager peut avoir un nombre différent de bagages par vol : Il est nécessaire de créer une classe association.
- Le nombre de passagers accédant au vol ne peut pas dépasser la capacité maximale du modèle d'avion.
- L'identifiant d'une porte d'embarquement n'est pas unique. Le local key de la porte est son identifiant.

Inès ABBACHE
Leonardoken D'ALESSIO
Hélène GAY
Zhentao XU

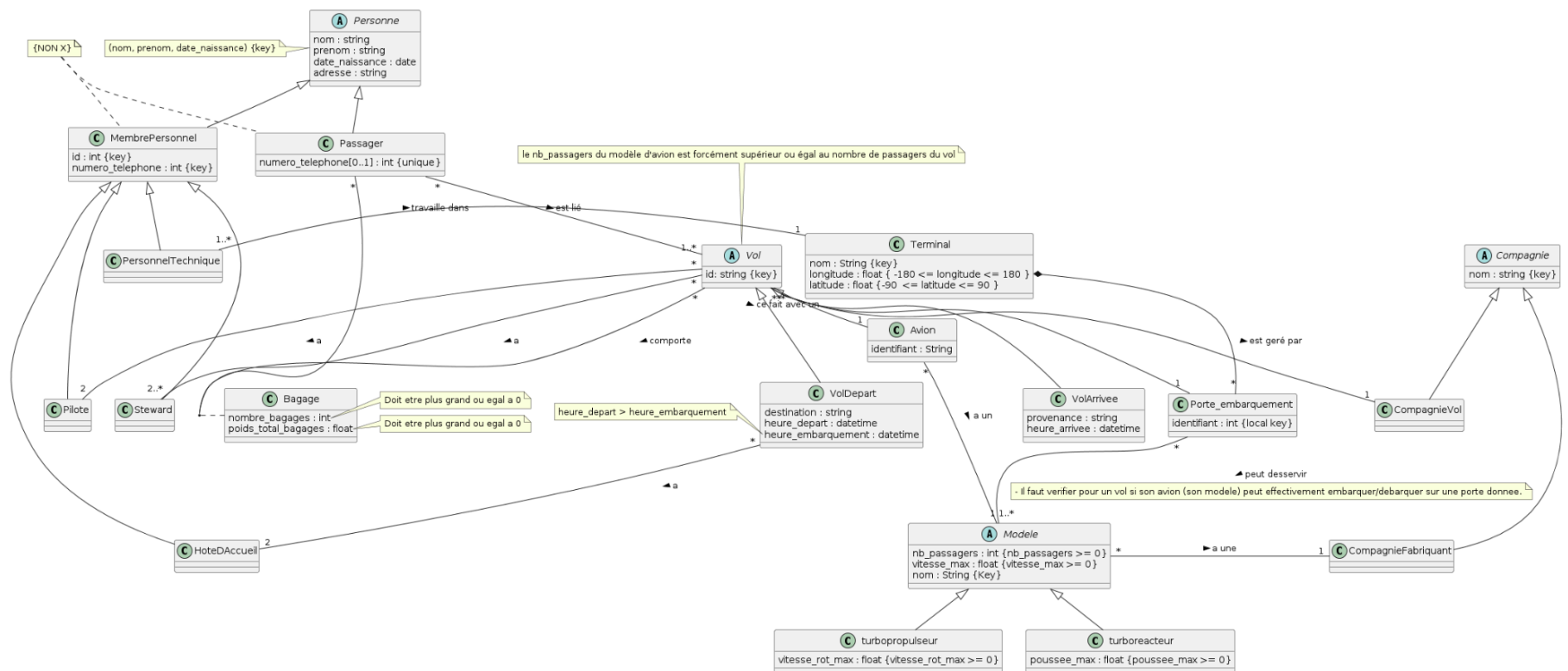
- La localisation est représentée par deux attributs de type float (longitude et latitude).
- L'identifiant d'un membre du personnel est unique (il sera donc la clé primaire).
- Le personnel technique travaille dans un terminal donné. Il y a au moins un personnel technique dans un terminal.
- La clé de Personne est nom, prénom et date de naissance.

Depuis le dernier TD:

- Le nom du modèle est une clé primaire.
- Un passager est lié à au moins un vol,
- Le numéro de téléphone de la classe des membres du personnel est une clé car il est unique et non null.
- L'héritage sur personne n'est pas exclusif car un membre du personnel pourrait être un passager.
- La classe membre_personnel est abstraite car un membre du personnel peut soit être un pilote, soit être un steward, un hôte d'accueil ou un membre du personnel technique.

MCD

UML



Inès ABBACHE
 Leonardoken D'ALESSIO
 Hélène GAY
 Zhentao XU

MLD

```
// tous les attributs seront supposes NOT NULL par default
// Il faut vérifier pour un vol si son avion (son modèle) peut effectivement embarquer/débarquer
// sur une porte donnée
// Le nombre de passagers accédant au vol ne peut pas dépasser la capacité maximale du
// modèle d'avion.
```

```
CompagnieVol(#nom : string)
CompagnieFabriquant(#nom : string)
// contraintes sur l'heritage par classe filles et mere abstraite
Intersection(Projection(CompagnieVol,nom), Projection(CompagnieFabriquant,nom)) = {}
```

```
Modele(#nom : string, nb_passagers : int, vitesse_max : float, vitesse_rot_max : float,
poussee_max : float,
compagnieFabriquant=>CompagnieFabriquant.nom, t:{"turbopropulseur","turboreacteur"})
avec {poussee_max NULLABLE, vitesse_rot_max NULLABLE, nb_passagers >= 0,
vitesse_max >= 0}
// contraintes sur l'heritage par classe mere abstraite
NOT (t = 'turbopropulseur' AND (poussee_max))
NOT (t = 'turboreacteur' AND (vitesse_rot_max))
poussee_max == NULL OR poussee_max >= 0
vitesse_rot_max == NULL OR vitesse_rot_max >= 0
```

```
Modele_Porte_Embarquement(#modele=>Modele,
#identifiant_porte=>PorteEmbarquement.identifiant,
#terminal_porte=>PorteEmbarquement.terminal)
//contrainte relation 1..N:N (Modele,PorteEmbarquement)
Projection(Modele, nom) = Projection(Modele_Porte_Embarquement, modele)
```

```
Avion(#identifiant : string, modele=>Modele.nom)
```

```
Passager(#nom : string, #prenom : string, #date_naissance : string, adresse : string,
numero_telephone : int)
avec {numero_telephone UNIQUE et NULLABLE}
```

Inès ABBACHE
Leonardoken D'ALESSIO
Hélène GAY
Zhentao XU

PersonnelTechnique(#id, numero_telephone : int, nom : string, prenom : string, date_naissance : date, adresse : string, terminal=>Terminal.nom)

avec {numero_telephone KEY, (nom, prenom, date_naissance) KEY}

//contrainte relation 1..N:1

Projection(PersonnelTechnique,terminal)=Projection(Terminal,nom)

Steward(#id, numero_telephone : int, nom : string, prenom : string, date_naissance : date, adresse : string)

avec {numero_telephone KEY, (nom, prenom, date_naissance) KEY}

Pilote(#id, numero_telephone : int, nom : string, prenom : string, date_naissance : date, adresse : string)

avec {numero_telephone KEY, (nom, prenom, date_naissance) KEY}

HoteDAccueil(#id, numero_telephone : int, nom : string, prenom : string, date_naissance : date, adresse : string)

avec {numero_telephone KEY, (nom, prenom, date_naissance) KEY}

//contrainte sur l'heritage (MemberPersonnel) par classes filles et classe mere abstraite

Intersection(Projection(PersonnelTechnique, id),Projection(Steward, id)) = {}

Intersection(Projection(PersonnelTechnique, id),Projection(Pilote, id)) = {}

Intersection(Projection(PersonnelTechnique, id),Projection(HoteDAccueil, id)) = {}

Intersection(Projection(Steward, id),Projection(Pilote, id)) = {}

Intersection(Projection(Steward, id),Projection(HoteDAccueil, id)) = {}

Intersection(Projection(Pilote, id),Projection(HoteDAccueil, id)) = {}

Terminal(#nom : string, longitude : float, latitude : float)

avec {-180 <= longitude <= 180, -90 <= latitude <= 90}

PorteEmbarquement(#identifiant, #terminal=>Terminal.nom)

Avion(#identifiant : string, modele=>Modele.nom)

// par reference

Vol(#id : string, compagnieVol=>CompagnieVol.nom, avion=>Avion.identifiant,

pilote_1=>Pilote.id, pilote_2=>Pilote.id, porteEmbarquement=>PorteEmbarquement.identifiant)

avec {pilote_1 <> pilote_2}

Inès ABBACHE

Leonardoken D'ALESSIO

Hélène GAY

Zhentao XU

```
Bagage(#vol=>Vol.id, #passager_nom=>Passager.nom,  
#passager_prenom=>Passager.prenom,  
#passager_date_naissance=>Passager.date_naissance, nombre_bagages : int,  
poids_total_bagages : float) avec{nombre_bagages >=0 AND poids_total_bagages>=0}  
//contrainte relation Bagage:Vol  
Projection(Bagage,vol) = Projection(Vol, id)  
  
//contrainte relation 1:1..N  
Projection(PorteEmbarquement,identifiant) = Projection(Vol,porteEmbarquement)  
  
Vol_Steward(#vol=>Vol, #steward=>Steward)  
//contrainte relation Steward:Vol_Steward  
Projection(Vol_Steward,steward) = Projection(Steward, id)  
Pour tous les vols dans la table (vol_i), il faut verifier que la cardinalite de  
(Restriction(Vol_Steward, vol = vol_i)) est superieure ou egale a 2.  
  
VolDepart(#id=>Vol, destination : string, heure_depart : datetime, heure_embarquement :  
datetime, hoteDAccueil_1=>HoteDAccueil.id,  
hoteDAccueil_2=>HoteDAccueil.id)avec{hoteDAccueil_1<>hoteDAccueil_2 AND  
heure_depart>heure_embarquement}  
  
VolArrive(#id=>Vol, provenance : string, heure_arrivee : datetime)  
// contraintes pour reference et classe mere abstraite  
Intersection(Projection(VolDepart, id),Projection(VolArrive, id)) = {}  
Projection(Vol,id) = Union (Projection(VolDepart,id), Projection(VolArrive,id))
```

Lien vers PlantUML

//www.plantuml.com/plantuml/png/bLNDRXit43w7Jp7GnKaGoKvEpl8nujOeMXHr0oGe2XI5GEs
EP29SSiCbrHlktwT-3hzOPtP4CRkh7dhIQkQRZz_ySFYkZph4r5gbQglx7ZvYQ8tJzau103ZVGW
KVOZ1k1tUVyA_xmMozsvcO6YJdrdeTuGwMBx-vW3TIYu__iEqGekCaOldw0XWZ7i6_PhYwps
BU-n1nZUR6Xyy9MtHHP9c6FitMQ1U9ppXYixxM5bXX2R_Q6-_c4JaX1kfD7uEk8qZ4pxv1YuVS
D-kER7g7eS-dZ6oNR-52ajkRY7sFwrP_AVbzRHJelA8_ZVJ653Q-2xvBjiSKxaRSmSSvVt786LE
oW9JRaQhpYG8cCejnHI6TRpkzSmQ_rvjWFYVD9M3th6w_UaiUqrJGVpqhX7K-8aJVWTyolcO
1nyLj4hl-UVo7xCsPvgO1zX6sFjIF3znrw5Ft-10CfW0-mUF3JblGYOkv2HI4C2BRqpde6k2pPrBV
4yWxXwHNUY6ALHTmzB_HD7fE6-9rDPAT3KetWOj-XEKCBMuZfvZPXpoV4k4B4E8NC2PyTY
XHqQaE8uYSK-QHiXx6JoBEp_OOQ9GnS8LQ7S6b5eC_ErVoiOveiRlr3aTnvLOPovAvasBu4F5
F7HgsPCmFXBY-gkk4nfAPiydHMTb7h6yTJOKmPU-X3sIRYldzOhLw-KUMb8PGkUpgE4JNsEd
0GzPW7utJqNXNQdsBAU2w4Omqq99eM1_jeJOoyumwLKwfg7ZUqRNJC9SnHfMb5-LK8PezSY
oxuFJhkyLlv5inOdA9WM3nVB9UBbujovTXQ3TU0gldu02I6j6GmB9VyKyfTGKsHI2wdg5UBtFOA
zCovMZrVvFOV_RaDXM8q3wU8IFfyMlrb30lZYgVmMe5t-eTVKIXv0yNxObSvVLcC1pshNhlUT
Fqw-YZjiKbcoltDPZTxT1OuQggCPC2kFOc0iQ0qDdKmover_0QmM6Bm4Lfvsd2iOZ_ophVGBd
7qnKqkQVZMP2zcrzdU9u7O5xq-GMbzd2VGVe4Hv8AIDPU6ykBLxkUcpedbE7C7F92KcY7AK
B3op3i3S-cLCyxiCPzgkYm7oriDO6EwwhpAK1ol9-XDq0gGRRzK_v1-PdGepRBTRHx4L1bXFE
6pnyGJzm8SZGDvmnha0Tz2rAkgTUiGeq1xFgArPGy3b1GS3bSSh5A-WAQgJSJ2JzM0ikd_pc
x-Lo-fBFVvbXLYhB4-yfjKfDtWH1ogq-wSiR_QHpT9TF8-OZF5Is2UAuU2RsAPsYvmfk-UsE10Vd
7U2NsgRURE7iN4PcSNVpomtyThy0tSFDAtL3Modp3tycfBmoHMXPviLKTh5IcKL6-331PRDQG
ukLP-2_kq1Wygftw9hKsdy1

Inès ABBACHE
Leonardoken D'ALESSIO
Hélène GAY
Zhentao XU