L'ALGEBRE RELATIONNELLE

L'expression algébrique

L'expression algébrique inclut, en plus des tables relationnelles, d'autres paramètres constitués de formules logiques construites à partir de :

constantes

attributs des tables

opérateur de comparaison (=, ≠, <, ≤, >, ≥)

connecteurs logiques (¬,∨,∧)

$$\Pi_{\textbf{numAvion, localisation}} (\sigma_{\textbf{localisation} \neq \textbf{'Nice'} \wedge \textbf{capacite} > \textbf{200}} (\text{Avion}))$$

L'algèbre relationnel

L'algèbre relationnelle (AR) est un système formel constitué d'opérations sur les relations.

Les opérations de l'AR satisfont la propriété de fermeture car, appliquées à une ou plusieurs relations, elles produisent une nouvelle relation.

Utilisation : démontrer l'équivalence de requêtes + écrire des optimiseurs de requêtes

Classification des opérateurs

Les opérations de base qui constituent un ensemble minimal d'opérations dans le sens où aucune d'entre elles ne peut s'écrire par combinaison des autres : union, différence, produit cartésien, projection, sélection, renommage.

Les opérations dérivées qui peuvent être exprimées à partir des opérations de base : intersection, théta-jointure, jointure naturelle et division.

Les opérations de groupage et agrégation qui constituent une extension des opérations de base sans pouvoir être exprimées à l'aide de celles-ci : compte, somme, min, max, moyenne...

L'AR comme langage de requêtes

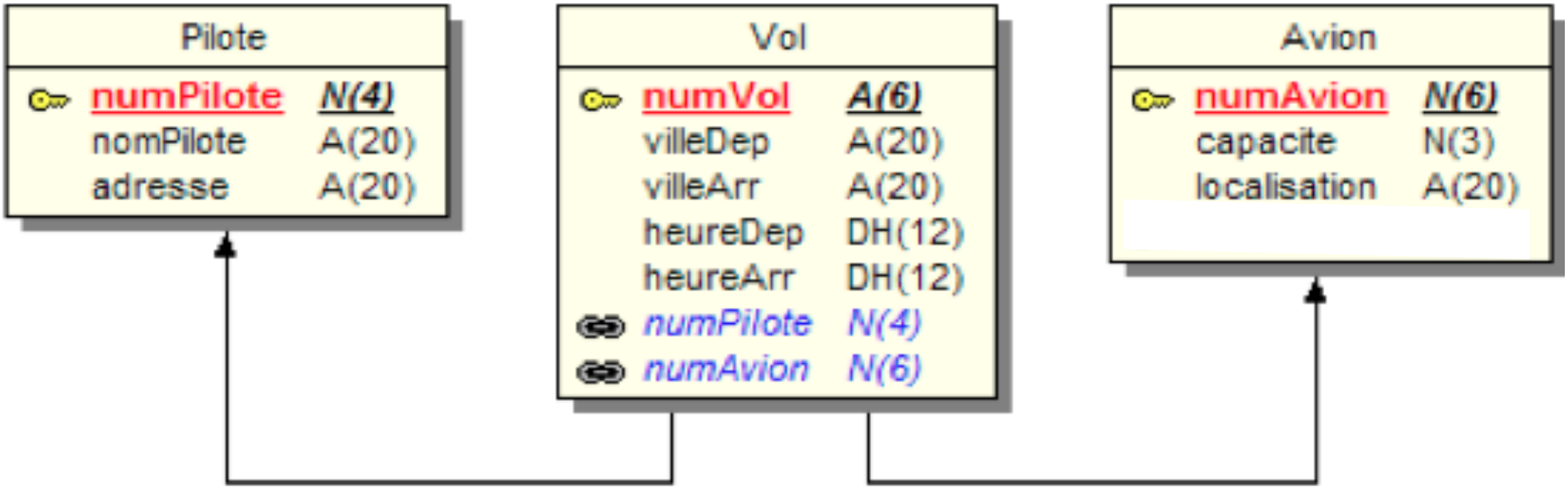
L'apprentissage de l'AR sert d'algorithmique préalable au codage des requêtes en SQL et permet d'améliorer l'écriture des requêtes.

$$\Pi_{\textbf{numAvion, localisation}} (\sigma_{\textbf{localisation} \neq \textbf{'Nice'} \wedge \textbf{capacite} > \textbf{200}} (\text{Avion}))$$

```
Select numAvion,localisation from Avion where localisation<>'nice' and capacite>200
```

Formuler en algèbre relationnelle une requête relative à une base de données relationnelle consiste à écrire une expression algébrique en composant des opérations de l'algèbre relationnelle ; chaque opération étant appliquée à des tables de la base de données.

Fil rouge du cours



Les opérateurs de base : la projection $\Pi_{a1,...,an}(R)$

$$\{u \mid \exists v \in \mathcal{R},\ \forall i \in \{1,\ldots,n\},\ u_{A_i} = v_{A_i}\}$$

Relation obtenue à partir de R en ne gardant des éléments de R que les colonnes a1,...,an.

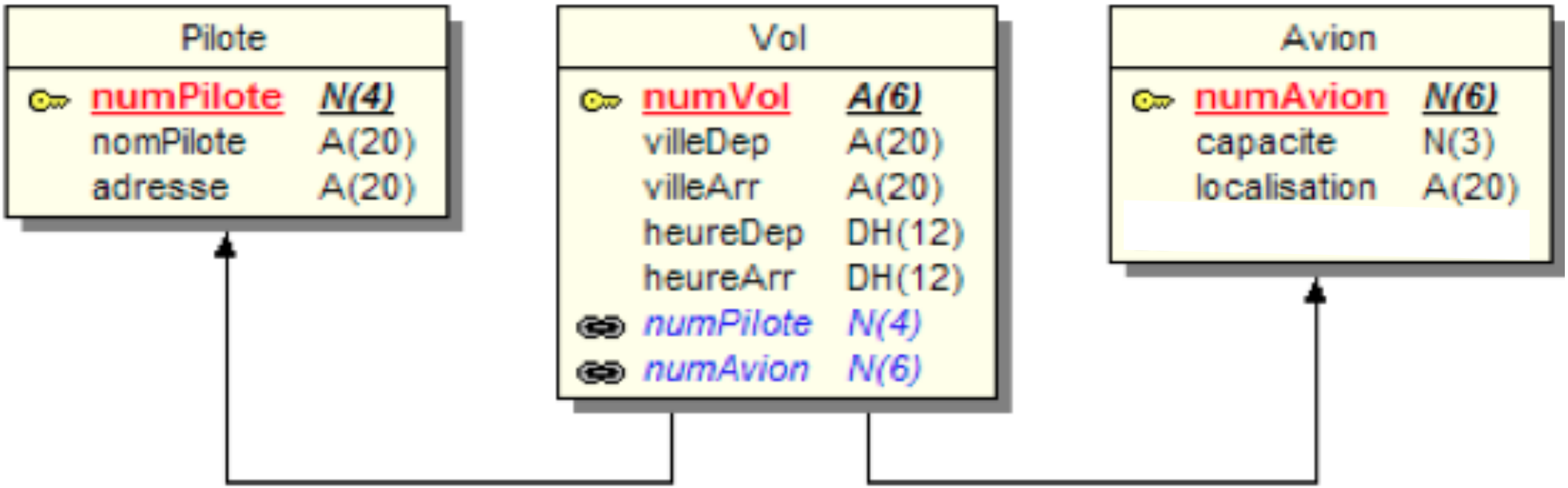
Attention : notion ensembliste donc pas de doublons ce qui explique la présence que d'une seule occurrence de (c,d) dans la projection.

r		
A	**B**	**C**
a	b	c
d	a	f
c	b	d
c	a	d

$\pi_{A, C}(r)$

A	**C**
a	c
d	f
c	d

Exemple de projection

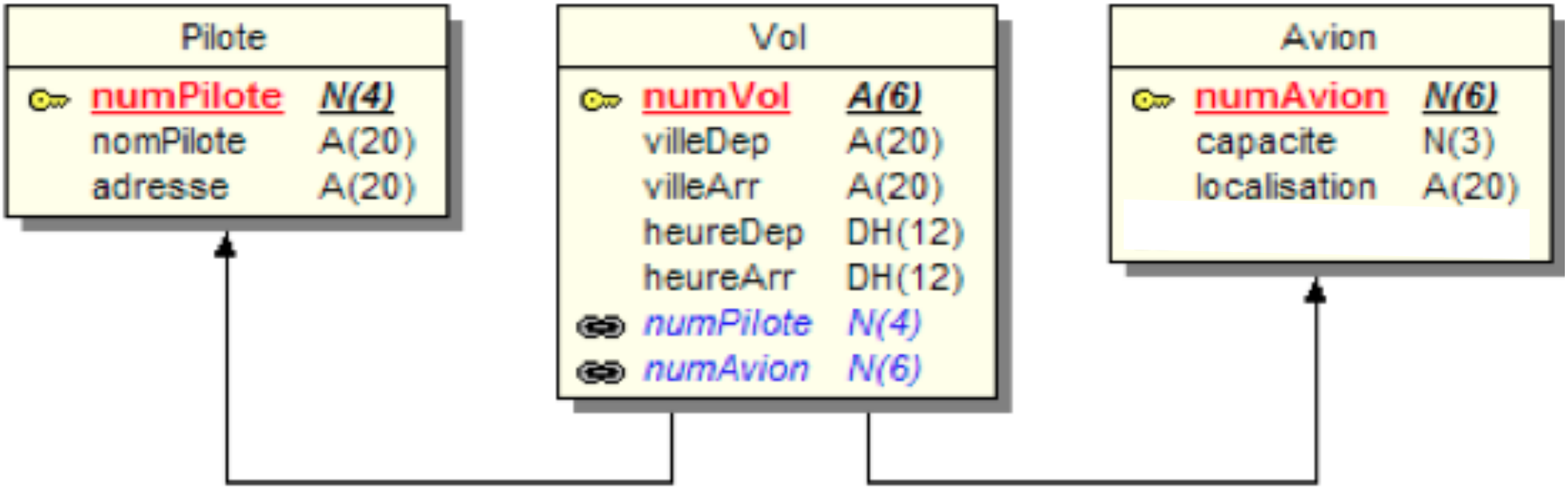


Quelles sont les différentes villes de départ de la table vol ?

Π_{villeDep} (Vol)

Quels sont les noms et adresses des pilotes ?

Exemple de projection



Quelles sont les différentes villes de départ de la table vol ?

$\Pi_{\textbf{villeDep}}$ (Vol)

Quels sont les noms et adresses des pilotes ?

$\Pi_{\textbf{nomPilote,adresse}}$ (Pilote)

Les opérations de base : la sélection σ_C (R)

Ensemble des éléments de R qui satisfont la condition de sélection (dite qualification) C.

$$\{u | u \in \mathcal{R} \wedge C(u)\}$$

Une qualification est une formule logique construite à partir des règles suivantes

- (Ai Op Aj) ou (Ai Op constante) avec Op $\in \{=, \neq, <, \leq, >, \geq\}$ (Ai,Aj des attributs de R) est une qualification,

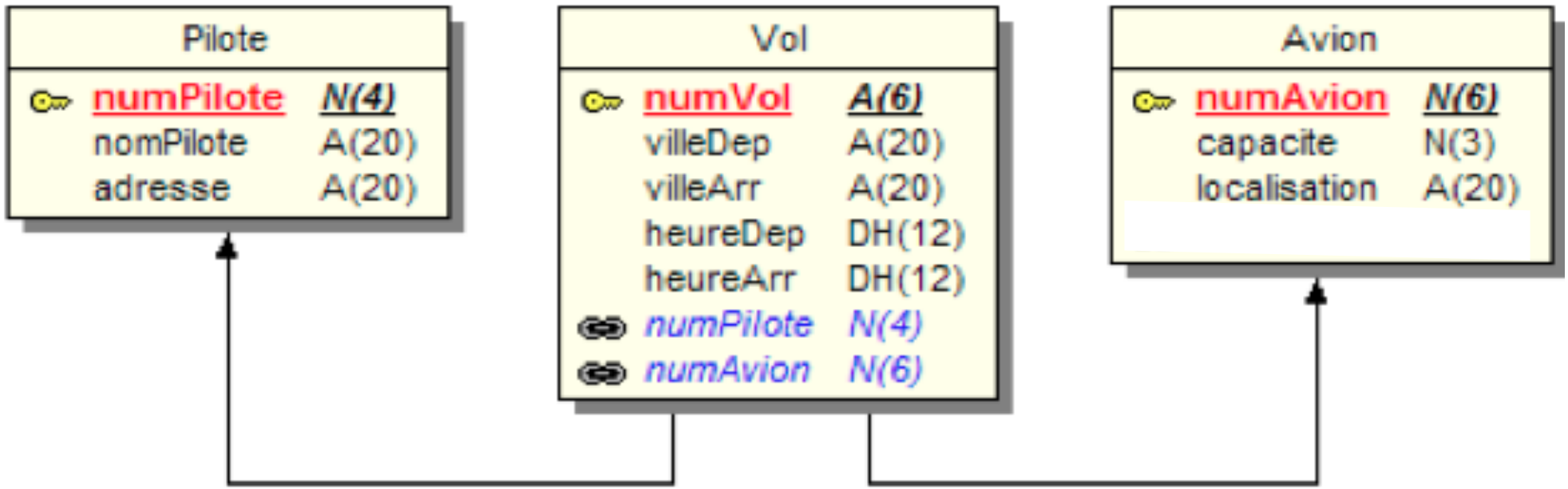
- si Q1 et Q2 sont des qualifications, alors (Q1), $\neg$ Q1, Q1 $\wedge$ Q2, Q1 $\vee$ Q2 sont des qualifications.

r		
A	**B**	**C**
a	b	c
d	a	f
c	b	d

$\sigma_{B = \text{'b'}}(r)$		
A	**B**	**C**
a	b	c
c	b	d

Les opérations de base : la sélection σ_C (R)

Exemple : quels sont les avions (numéro et localisation) non localisés à Nice ayant une capacité supérieure à 200 places.



Les opérations de base : la sélection σ_C (R)

Exemple : quels sont les avions (numéro et localisation) non localisés à Nice ayant une capacité supérieure à 200 places.

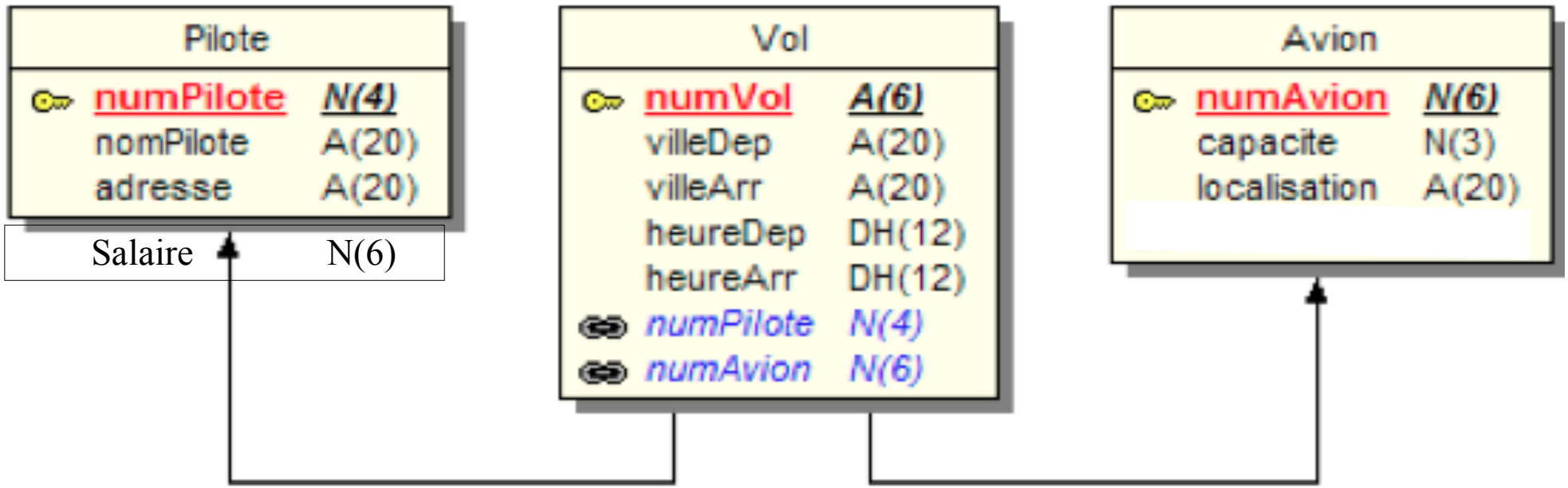
$$\Pi_{\textbf{numAvion, localisation}} (\sigma_{\textbf{localisation} \neq \textbf{'Nice'} \wedge \textbf{capacite} > \textbf{200}} (\text{Avion}))$$

Remarque : en cas d'ambiguïté sur le nom d'un attribut un renommage et/ou préfixage par le nom de la relation peut s'avérer indispensable.

Les opérations de base : la sélection σ_C (R)

Exemple : Rajoutons une colonne Salaire dans la table Pilote.

Quels sont les noms des pilotes domiciliés à Paris dont le salaire est supérieur à 3000



Les opérations de base : la sélection σ_C (R)

Exemple : Quels sont les noms des pilotes domiciliés à Paris dont le salaire est supérieur à 3000

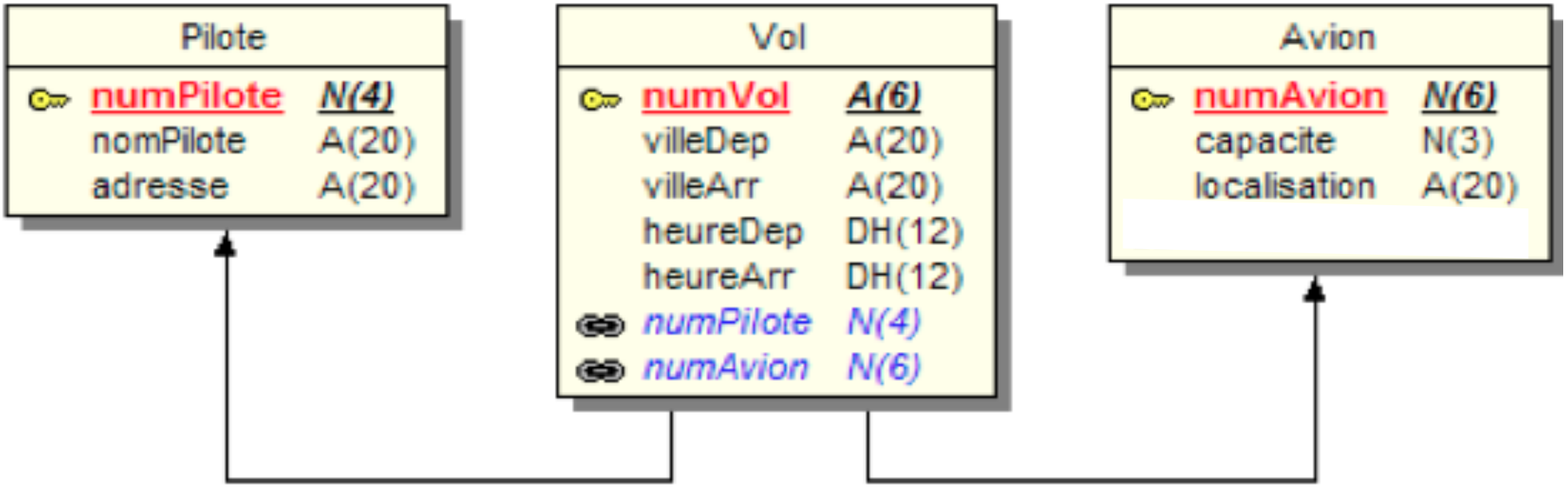
$$\Pi_{\textbf{nompilote}} (\sigma_{\textbf{adresse= 'Paris' } \wedge \textbf{ salaire > 3000}} (\text{Pilote}))$$

Les opérations ensemblistes de base : l'union R∪S

Ensemble des éléments apparaissant dans R ou dans S.

$$\{u | u \in \mathcal{R} \vee u \in S\}$$

Exemple : Quels sont les avions (numéro) localisés à Nice ou dont la capacité est inférieure à 350

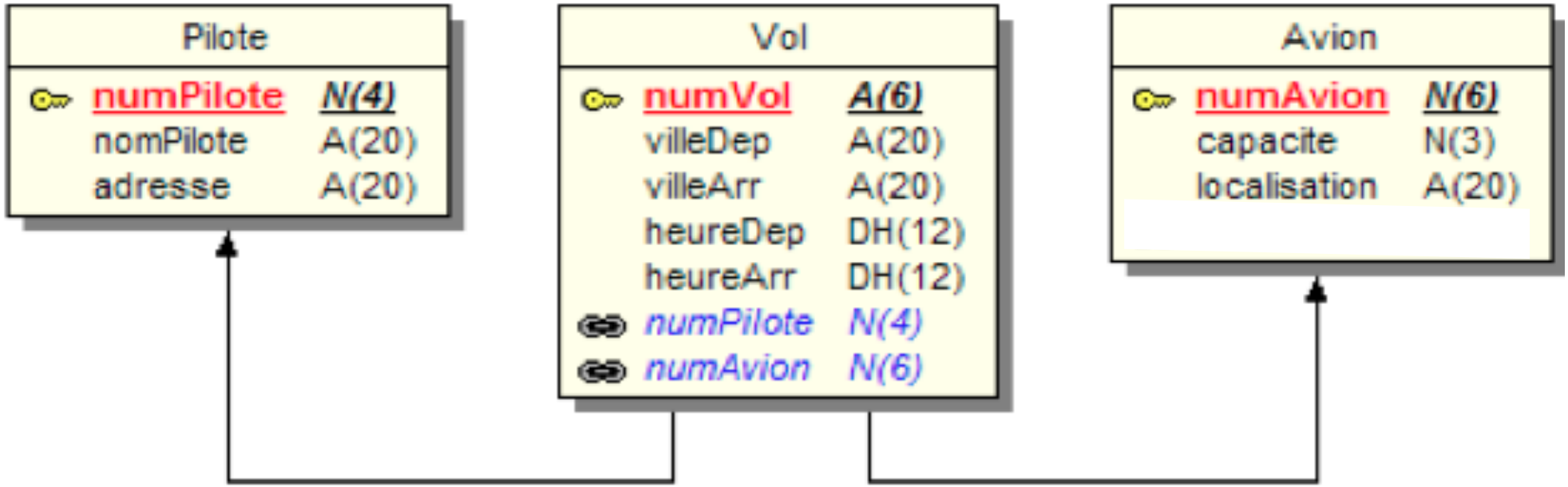


Les opérations ensemblistes de base : l'union R∪S

Exemple : Quels sont les avions (numéro) localisés à Nice ou dont la capacité est inférieure à 350

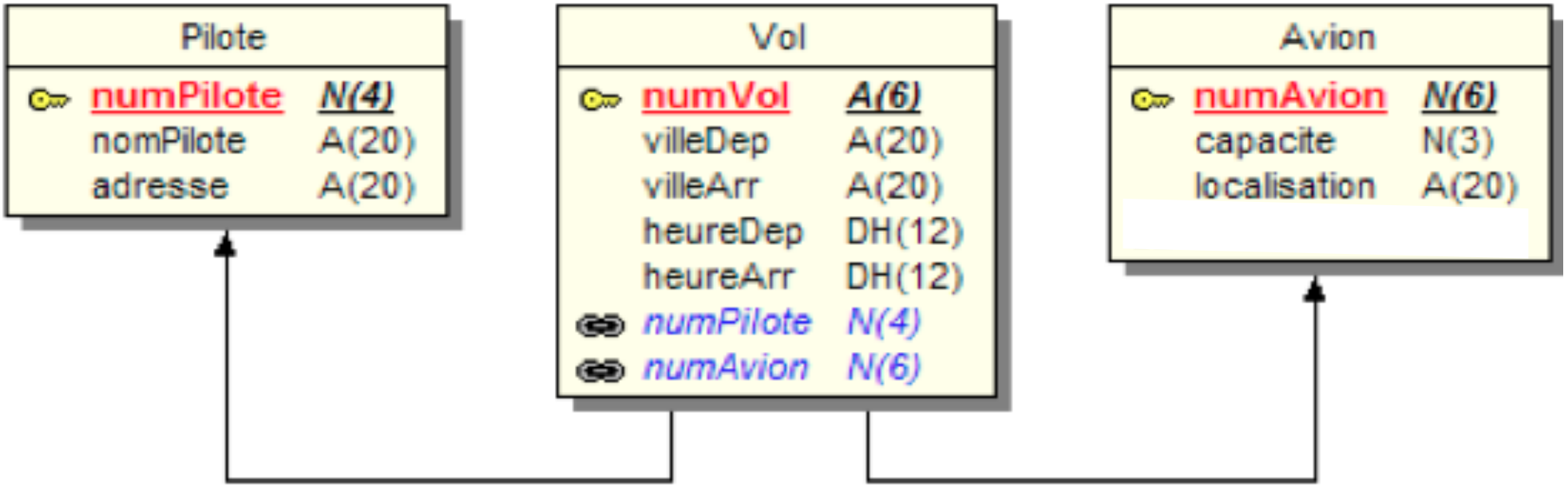
$$\Pi_{\text{numavion}}(\sigma_{\text{localisation= 'Nice'}}(\text{Avion})) \quad \cup$$

$$\Pi_{\text{numavion}}(\sigma_{\text{capacité<350}}(\text{Avion}))$$



Les opérations ensemblistes de base : l'union R∪S

Exemple : Quels sont les villes de départ, soit des vols à destination de Melbourne, soit des vols à destination de Sydney mais dont l'heure d'arrivée est prévue avant 14H.

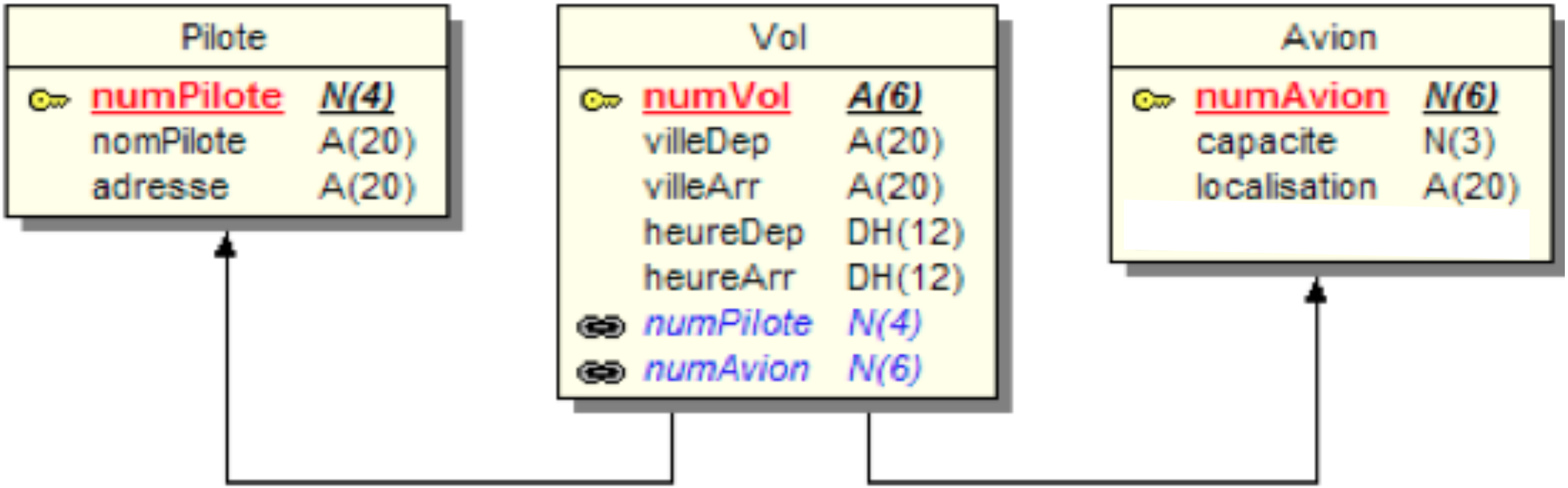


Les opérations ensemblistes de base : l'union R∪S

Exemple : Quels sont les villes de départ, soit des vols à destination de Melbourne, soit des vols à destination de Sydney mais dont l'heure d'arrivée est prévue avant 14H.

$$\Pi_{\text{villeDep}}(\sigma_{\text{villeArr= 'Melbourne'}}(\text{Vol})) \quad \cup$$

$$\Pi_{\text{villeDep}}(\sigma_{\text{villeArr='Sydney'} \wedge \text{heureArr<14:00}}(\text{Vol}))$$



Les opérations ensemblistes de base : l'union R∪S

Remarque : il est conseillé que les tables R et S soient compatibles, c'est-à-dire :

(1) Cohérence entre le nombre d'attributs :

Contre exemple : R(a1,a2) et S(a3) → R∪S ?

Solution : $\Pi_{a1}(R) \cup \Pi_{a3}(S)$

(2) Même groupe type du domaine (data type group) respectif :

Contre exemple : $\Pi_{a1}(R) \cup \Pi_{a2}(S)$ avec a1 caractère et a2 numérique !!!

En effet lors du passage vers ORACLE ces deux contraintes doivent être respectées.

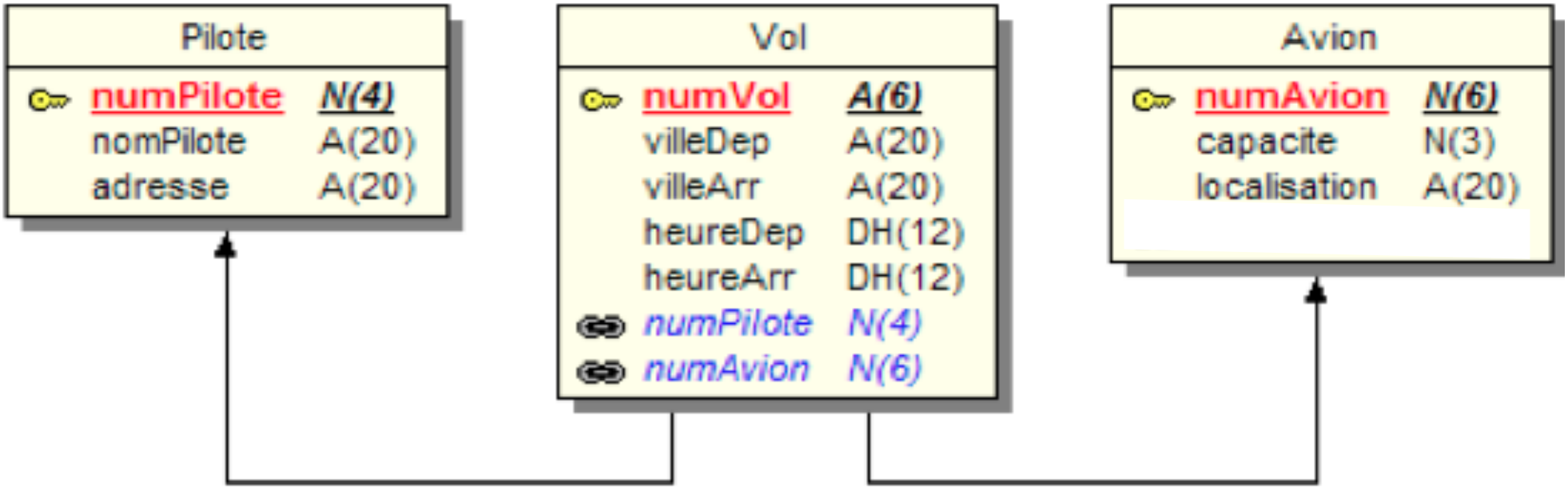
Même remarque pour l'intersection et la différence.

Les opérations ensemblistes de base : la différence R \ S

Ensemble des éléments apparaissant dans R mais pas dans S.

$$\{u | u \in \mathcal{R} \wedge u \notin S\}$$

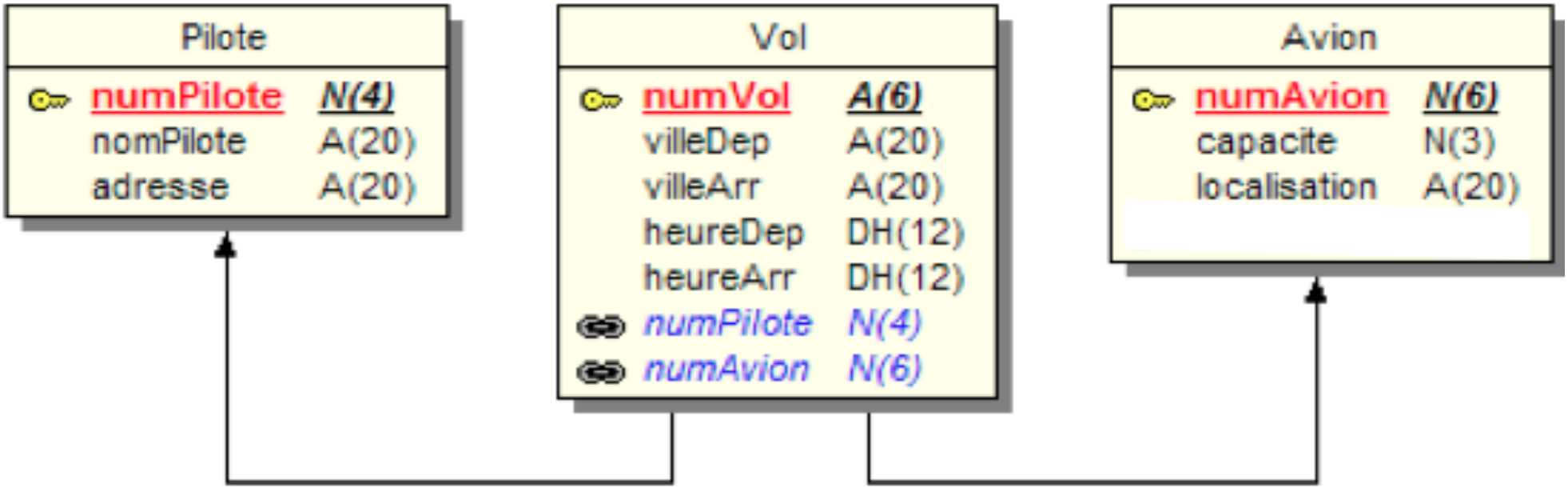
Très pratique pour les question avec « aucun » : quels sont les pilotes (numéro) qui n'assurent aucun vol ?



Les opérations ensemblistes de base : la différence R \ S

Quels sont les pilotes (numéro) qui n'assurent aucun vol ?

$$\Pi_{\text{numPilote}} (\text{Pilote}) \setminus \Pi_{\text{numPilote}} (\text{Vol})$$



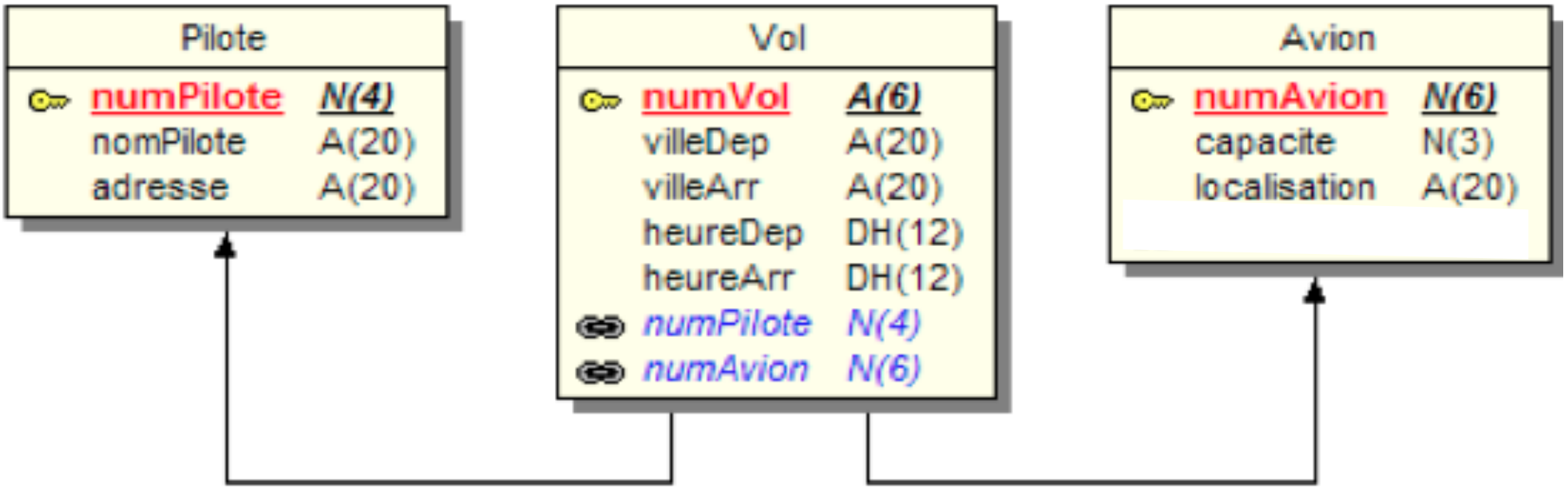
Les opérations ensemblistes de base : la différence R \ S

Donnez les numéros de pilotes qui ont fait au moins un vol mais qui n'ont jamais utilisé l'avion numéro 320.

On serait tenter de faire :

$$\Pi_{\text{numpilote}} (\sigma_{\text{numavion} \neq 320} (\text{Vol}))$$

Mais pourquoi c'est faux ?



Les opérations ensemblistes de base : la différence R \ S

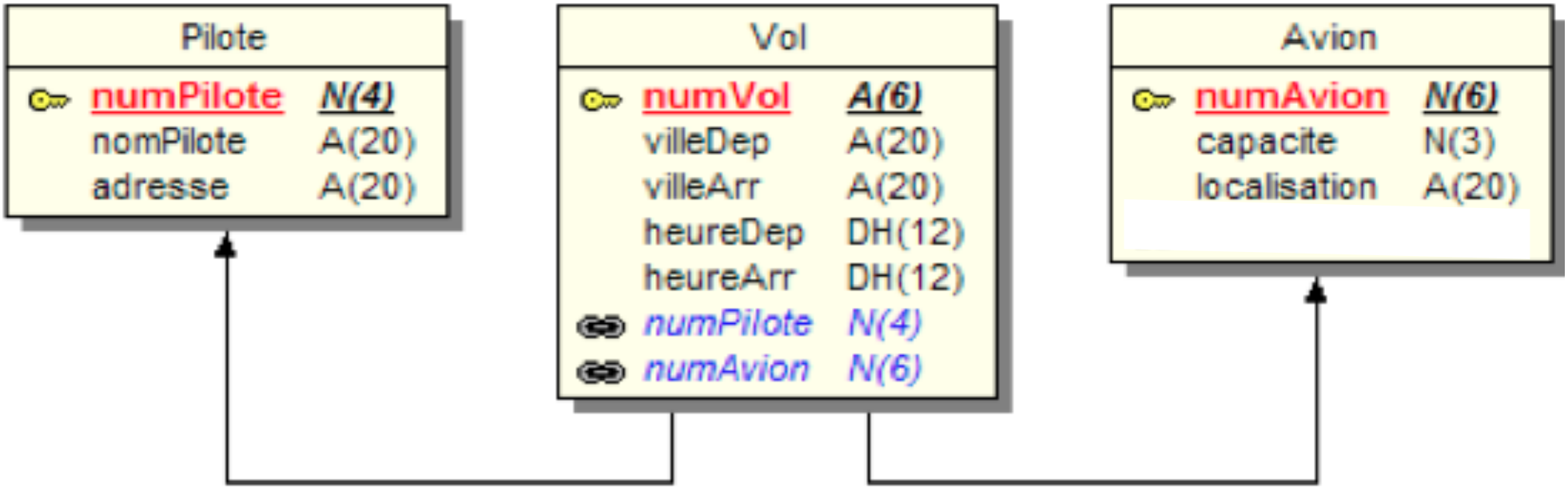
$$\Pi_{\text{numpilote}} (\sigma_{\text{numavion} \neq 320} (\text{Vol}))$$

Supposons que le pilote 1 a fait le vol AF1 dans l'avion 320 ainsi qu'un vol AF2 dans l'avion 400

On a donc deux lignes dans la table Vol avec le pilote 1.

Notre requête va supprimer la première ligne car le numéro de l'avion est 320 mais elle va retenir la seconde ligne car le numéro de l'avion est différent de 320. Le pilote 1 sera donc affiché !

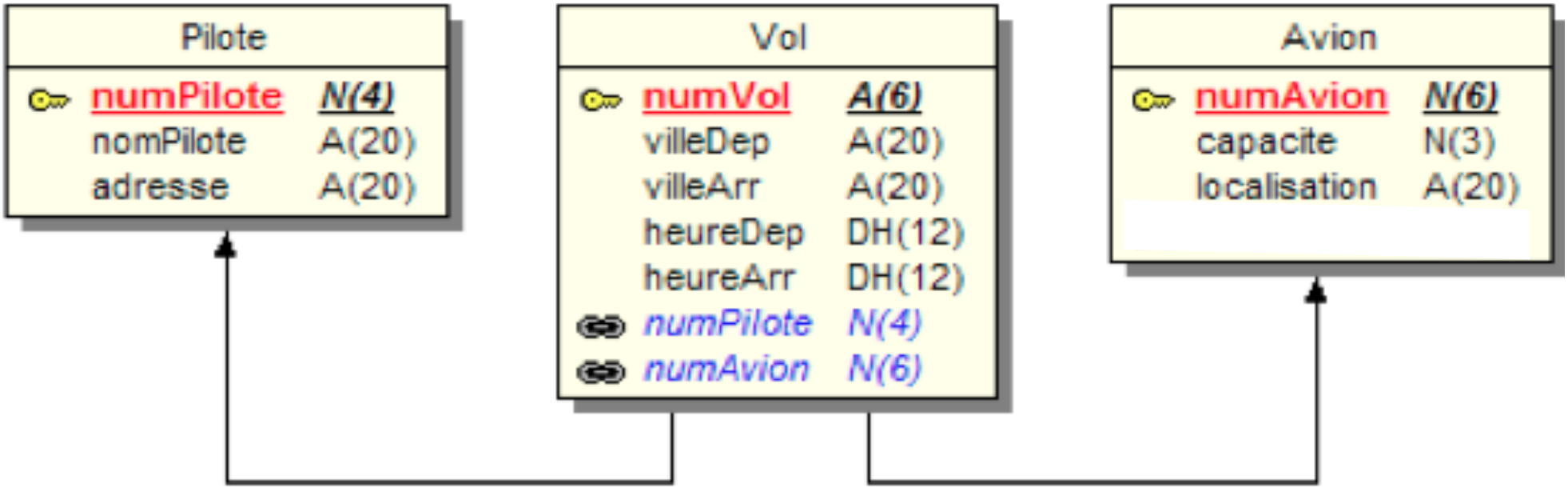
Corrigez cette requete et méfiez vous des question avec une négation dans l'énoncé !



Les opérations ensemblistes de base : la différence R \ S

Donnez les numéros de pilotes qui ont fait au moins un vol mais qui n'ont jamais utilisé l'avion numéro 320.

$$\Pi_{\text{numpilote}}(\text{Vol}) \setminus \Pi_{\text{numpilote}}(\sigma_{\text{numavion= 320}}(\text{Vol}))$$



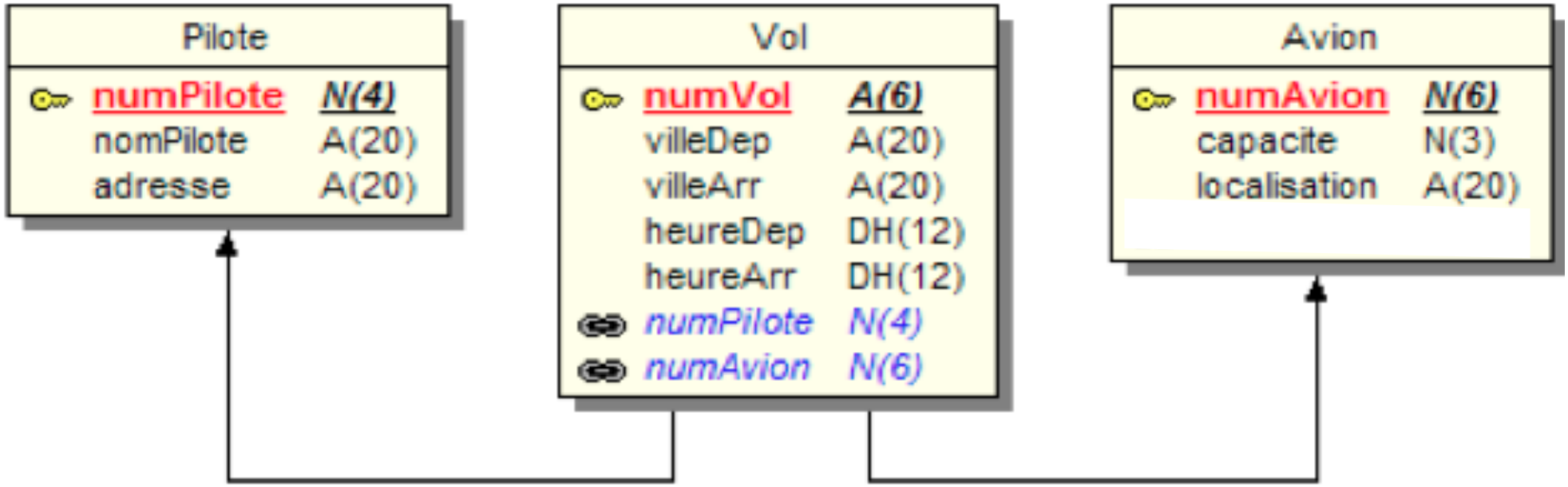
Les opérations dérivées : l'intersection R∩S

Ensemble des éléments apparaissant à la fois dans R et dans S.

$$\{u | u \in \mathcal{R} \wedge u \in S\}$$

R ∩ s=r \ (r \ s) =s \ (s\ r)

Exemple : quels sont les heures de départ des vols à destination de Melbourne qui sont également des heures de départ de vols de Paris ou de Nice.

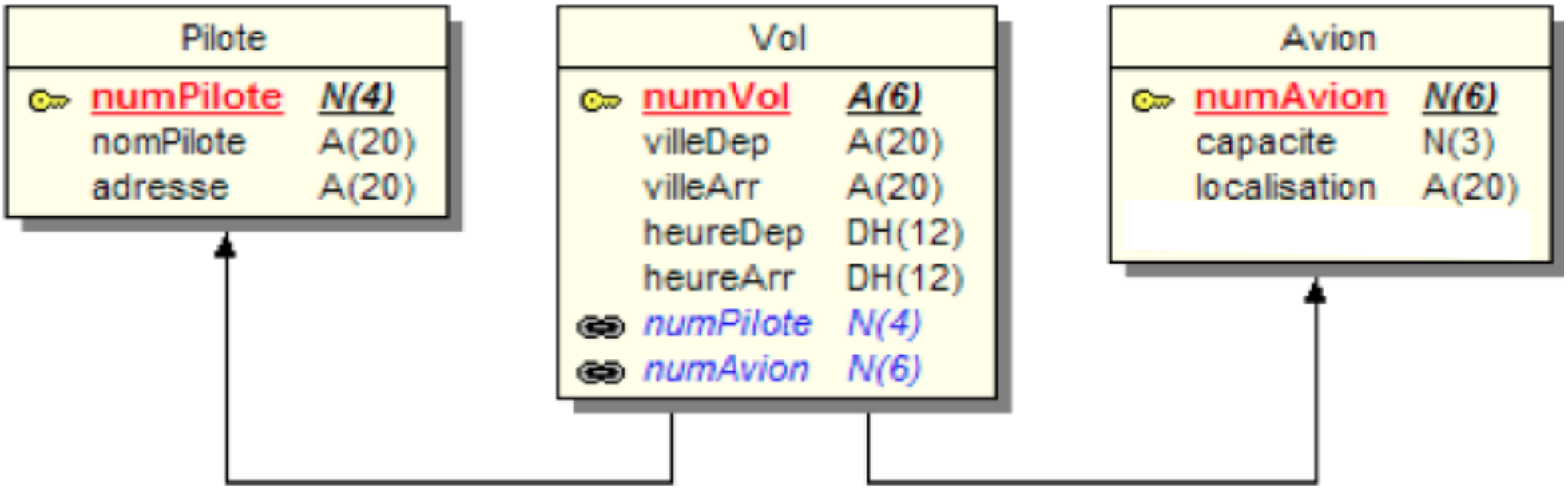


Les opérations dérivées : l'intersection R∩S

Exemple : quels sont les heures de départ des vols à destination de Melbourne qui sont également des heures de départ de vols de Paris ou de Nice.

$$\Pi_{\text{heureDep}}(\sigma_{\text{villeArr= 'Melbourne'}}(\text{Vol})) \quad \cap$$

$$\Pi_{\text{heureDep}}(\sigma_{\text{villeDep='Paris'} \vee \text{villeDep='Nice'}}(\text{Vol}))$$



Les opérations dérivées : l'intersection R∩S

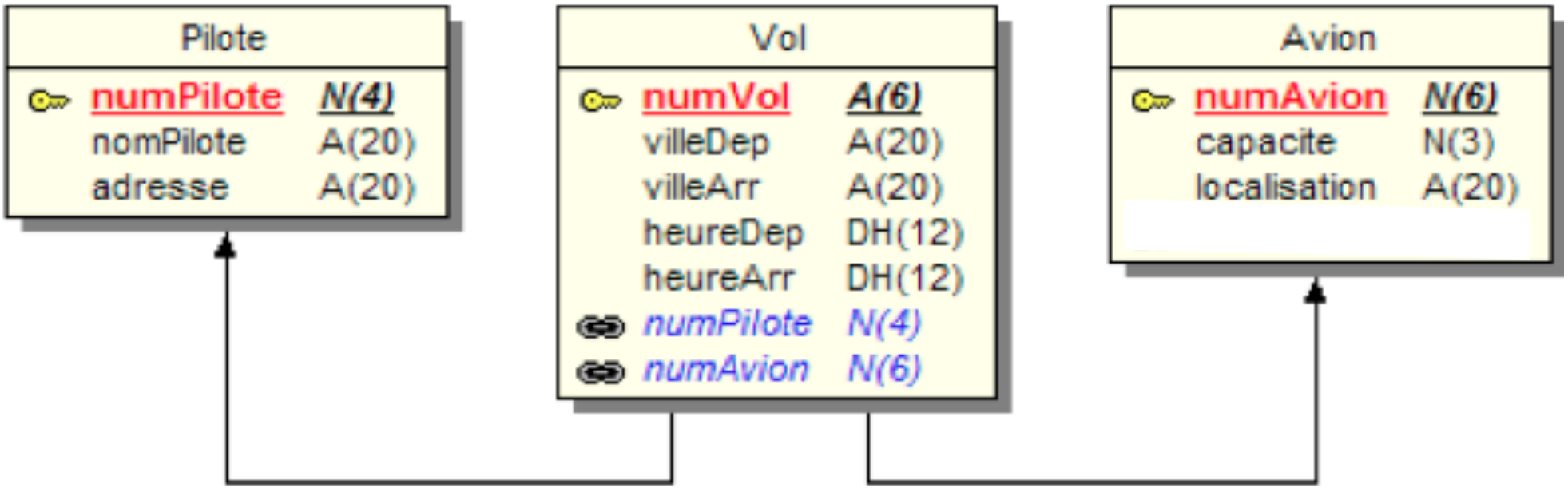
Supposons que vous avez déjà écrit les requêtes suivantes :

Req1 : liste des vols qui partent de Paris.

Req 2 : liste des vols qui arrivent à Marseille

Req 3 : liste des vols qui utilisent l'avion numéro 320

On souhaite en utilisant uniquement les requêtes 1, 2 et 3 obtenir la liste des vols qui partent de Paris et arrivent à Marseille en utilisant l'avion 320.



Les opérations dérivées : l'intersection R∩S

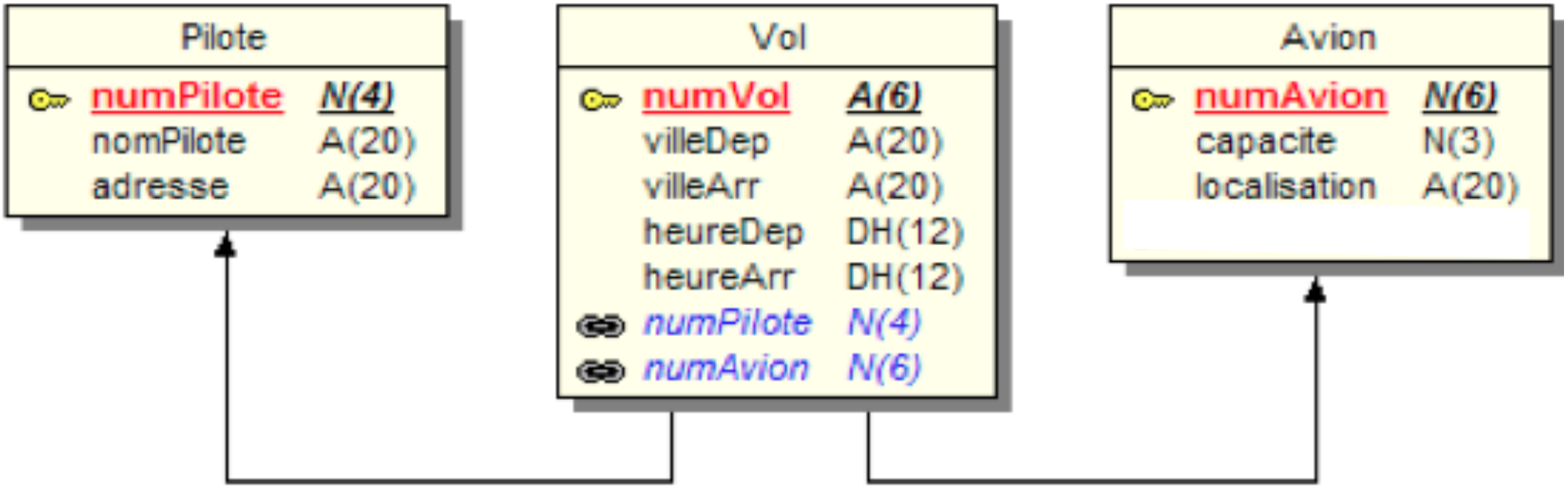
Req1 : numéros des vols qui partent de Paris.

Req 2 : numéros des vols qui arrivent à Marseille

Req 3 : numéros des vols qui utilisent l'avion numéro 320

On souhaite en utilisant uniquement les requêtes 1, 2 et 3 obtenir la liste des vols qui partent de Paris et arrivent à Marseille en utilisant l'avion 320.

Req1 ∩ Req1 ∩ Req1



Les opérations ensemblistes de base : le produit cartésien R × S

Ensemble des tuples constitués à partir d'un élément de R et d'un élément de S.

$$\{u | \exists v \in \mathcal{R},\ \exists w \in S,\ u = (v_1, \ldots, v_n, w_1, \ldots, w_m)\}$$

Attention au coût de cette opération !

r		
A	**B**	**C**
a	b	c
d	a	f
c	b	d

s		
A	**B**	**C**
b	g	a
d	a	f

r × s					
r.A	**r.B**	**r.C**	**s.A**	**s.B**	**s.C**
a	b	c	b	g	a
a	b	c	d	a	f
d	a	f	b	g	a
d	a	f	d	a	f
c	b	d	b	g	a
c	b	d	d	a	f

Les opérations dérivées : la jointure

La jointure sous toute ses formes est une opération essentielle pour les BDD relationnelles car elle permet une utilisation raisonnable du produit cartésien et permet d'assurer très simplement la navigation dans les BDD relationnelles.

La théta-jointure $R \bowtie_C S$

Constituée des n-uplets de $R \times S$ qui satisfont la condition C.

$R \bowtie_C S = \sigma_C (R \times S)$

r

A	B	C
1	2	3
4	5	6
7	8	9

s

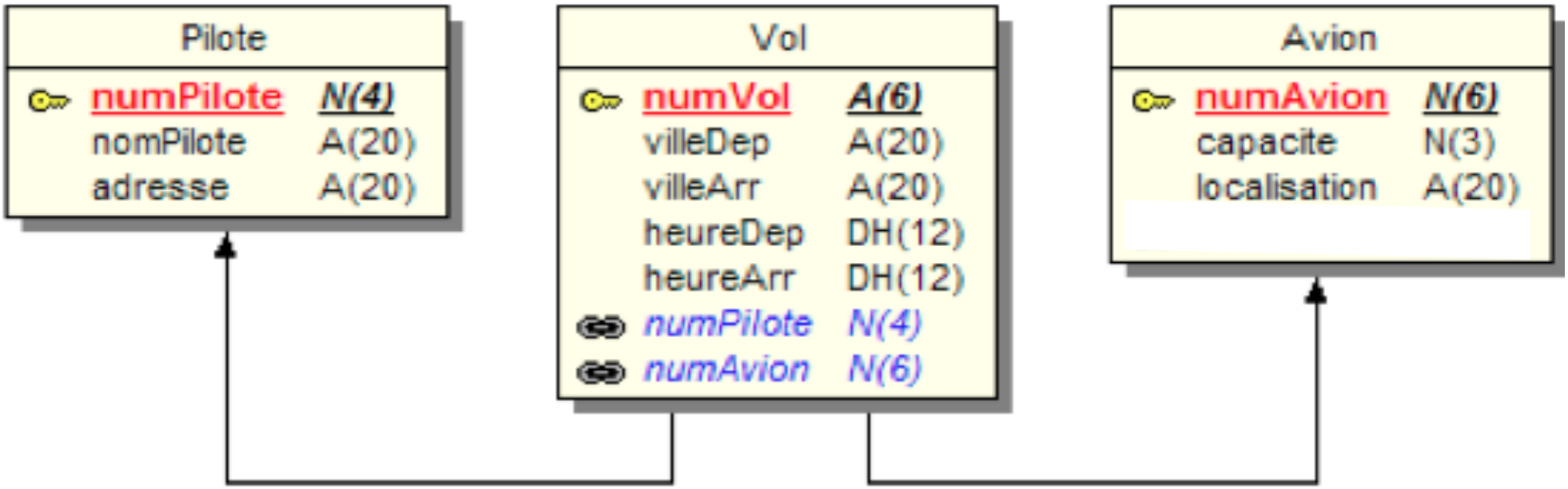
D	E
3	1
6	2

$r \bowtie_{B<D} s$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

La théta-jointure $R \bowtie_c S$

Exemple : quels sont les numéros des vols au départ de Londres et à destination de Paris dont l'heure d'arrivée prévue est antérieure à l'heure de départ de vol au départ de Paris et à destination de Melbourne.

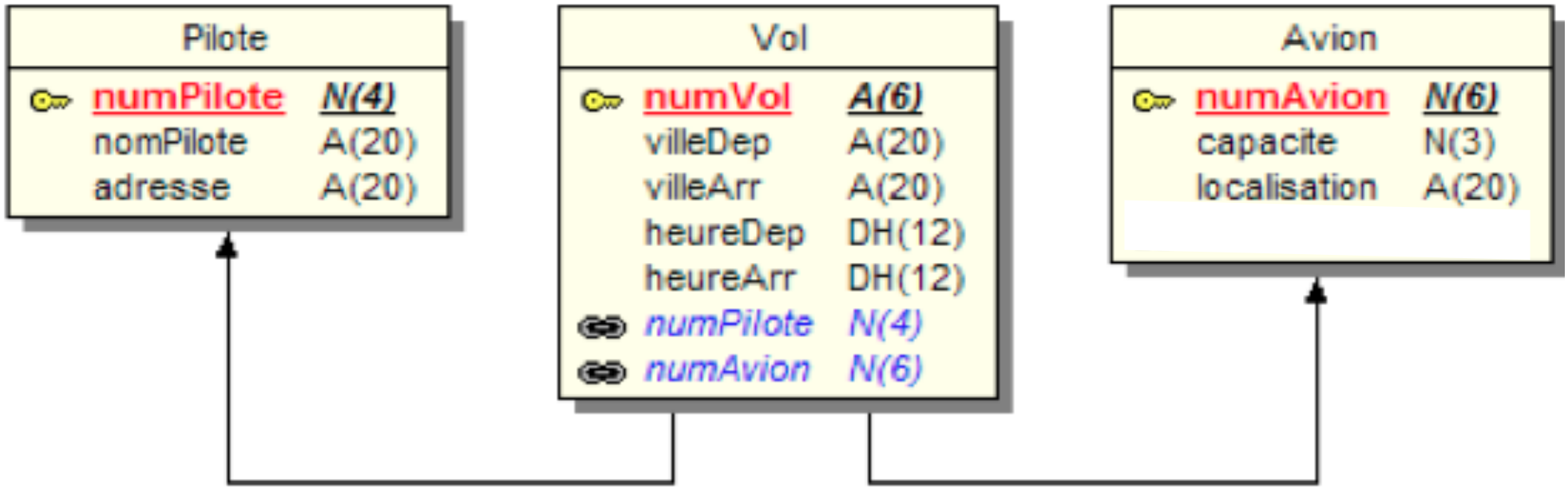


Indication : on doit utiliser une jointure de la table vol sur elle même

Problème : ambiguité des noms des attributs → renommage.

La théta-jointure $R \bowtie_c S$

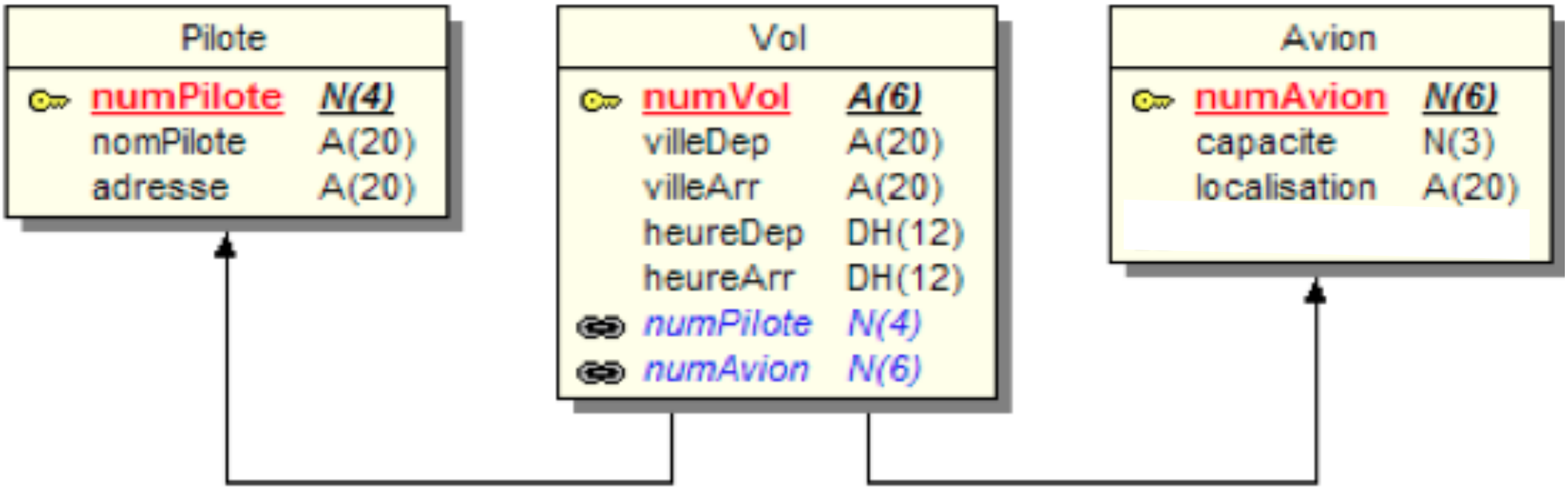
Exemple : quels sont les numéros des vols au départ de Londres et à destination de Paris dont l'heure d'arrivée prévue est antérieure à l'heure de départ de vol au départ de Paris et à destination de Melbourne.



Solution 1 : suffixer les noms de tables par des entiers → vol qui devient vol_1 et vol_2

La théta-jointure $R \bowtie_c S$

Exemple : quels sont les numéros des vols au départ de Londres et à destination de Paris dont l'heure d'arrivée prévue est antérieure à l'heure de départ de vol au départ de Paris et à destination de Melbourne.



Solution 1 : suffixer les noms de tables par des entiers → vol qui devient vol_1 et vol_2

$$\Pi_{numVol_1}\Big(\sigma_{villeDep_1=\text{'Londre'} \wedge villeArr_1=\text{'Paris'}}(Vol_1) \bowtie_{heureArr_1<heureDep_2} \sigma_{villeDep_2=\text{'Paris'} \wedge villeArr_2=\text{'Melbourne'}}(Vol_2)\Big)$$

Solution 2 : créer une opération de base dédiée au renommage → $\rho_{S(A1,\ldots,An)}(R)$

Il en existe deux formes :

$\rho_{S(A1,\ldots,An)}(R)$: relation obtenue en renommant R en S et ses attributs en A1, . . . , An.

$\rho_{S}(R)$: relation obtenue en renommant R en S. Les attributs sont accessibles via S.nomAttribut

Reprenons l'exemple : quels sont les numéros des vols au départ de Londres et à destination de Paris dont l'heure d'arrivée prévue est antérieure à l'heure de départ de vol au départ de Paris et à destination de Melbourne.