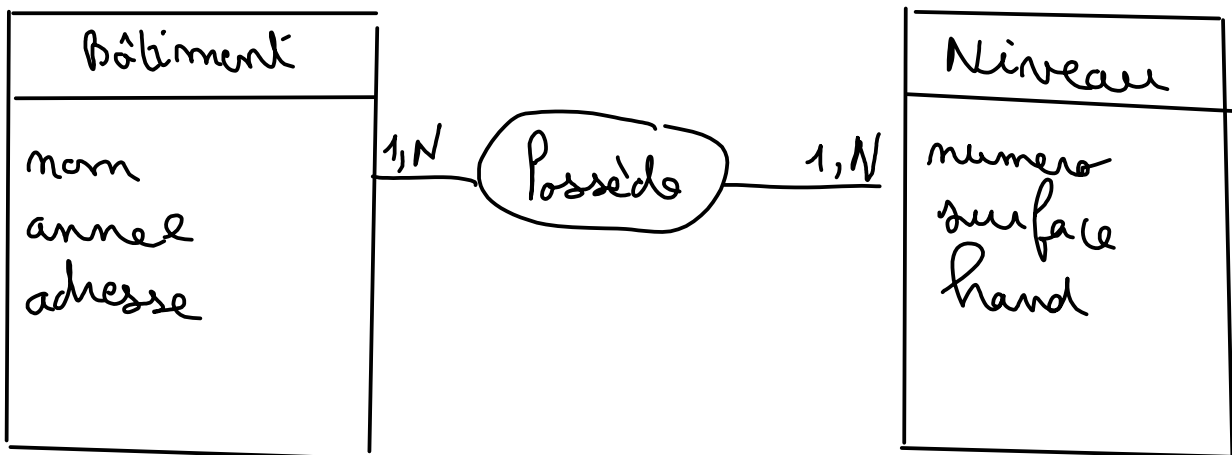


## Exo conception BDD

L'université souhaite gérer ses bâtiments et leur entretien. Modéliser les données en E/A, puis traduire en relationnel, à partir des spécifications suivantes :

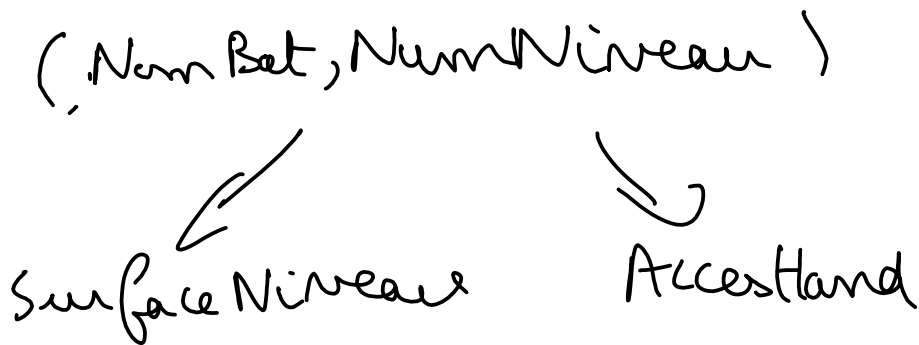
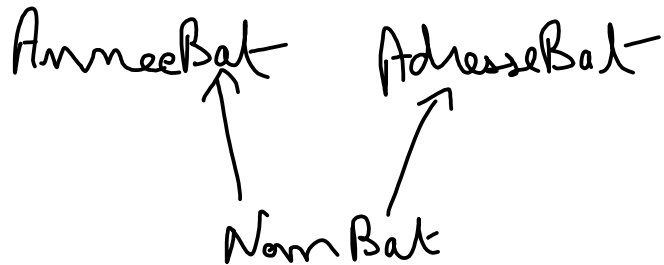
- L'université gère un ensemble de bâtiments, qui possèdent un nom unique. On souhaite stocker leur année de construction et leur adresse.
- Chaque bâtiment peut avoir plusieurs niveaux, repérés par leur numéro en partant du 0 pour le rez-de-chaussée. Chaque niveau a une surface donnée et un booléen permettant de savoir si l'étage est accessible aux personnes en fauteuil roulant ou pas.



1- Soit la liste des attributs suivante issue de ces spécifications :

Dresser l'inventaire des dépendances fonctionnelles entre les attributs suivants : NomBat, AnneeBat, AdresseBat, NumNiveau, SurfaceNiveau, AccesHand.

Schéma des dépendances fonctionnelles  
entre les différents attributs



la surface d'un niveau et son accès handicapé  
ne dépendent pas que de son numéro  
mais aussi du bâtiment auquel il se  
rattache.

## Correction en vert. :

- ci-dessus le cahier des charges avec des dépendances fonctionnelles
- ci-dessous les dépendances fonctionnelles qui apparaissent dans le cahier des charges

1- Soit la liste des attributs suivante issue de ces spécifications :

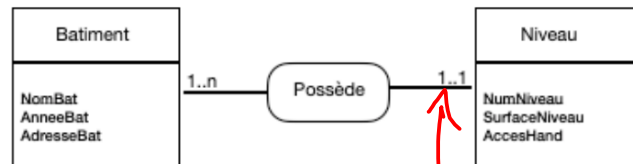
Dresser l'inventaire des dépendances fonctionnelles entre les attributs suivants : NomBat, AnnéeBat, AdresseBat, NumNiveau, SurfaceNiveau, AccesHand.

NomBat  $\rightarrow$  AnnéeBat, AdresseBat

NomBat, NumNiveau  $\rightarrow$  SurfaceNiveau, AccesHand

- On part du cahier des charges qu'on tire tout en dépendances fonctionnelles

2- Le schéma E/A suivant propose un extrait de modélisation en ce qui concerne les bâtiments et les étages. Traduisez ce schéma en relationnel, puis critiquez le résultat en vous appuyant sur les DF trouvées plus haut.



- Cardinalité fautive, un triplet (NumNiveau, SurfaceNiveau, AccesHand) peut être partagé par plusieurs bâtiments

→ PK  
 — FK

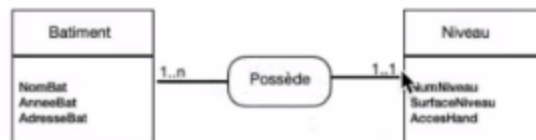
• Schéma relationnel :

Batiment ( NomBat : PK int ,  
 AnnéeBat : not null int ,  
 Adresse Bat : not null text )

Niveau ( NumNiveau : PK int ,  
~~NomBat : not null int - FK references Bat (NomBat)~~  
 Surface Niveau : not null int -  
 AccesHand : not null boolean )

Correction :

2- Le schéma E/A suivant propose un extrait de modélisation en ce qui concerne les bâtiments les étages. Traduisez ce schéma en relationnel, puis critiquez le résultat en vous appuyant sur DF trouvées plus haut.

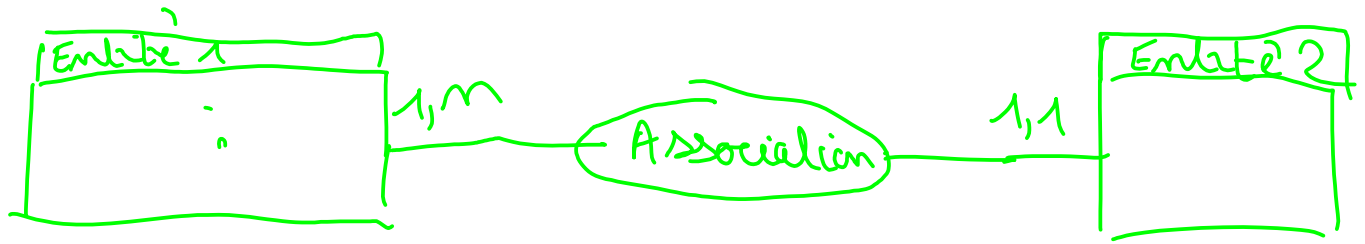


Batiment(NomBat, AnnéeBat, AdresseBat)  
 Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)

DF dans ce schéma relationnel :  
 NomBat -> AnnéeBat, AdresseBat  
 NumNiveau -> NomBat, SurfaceNiveau, AccesHand

=> cahier des charges non respecté, car le numéro de niveau n'est pas unique dans l'établissement (il dépend du bâtiment).

- Cardinalité Max



Schema E/A

⇓ Traduction en modèle relationnel

On importe une def primaire de la table traduisant l'entité 1 comme def étrangère dans la table traduisant l'entité 2

- L'entité Niveau ne peut être définie uniquement par un numéro Niveau

Le schéma association ci-dessus ne correspond pas à notre cahier des charges

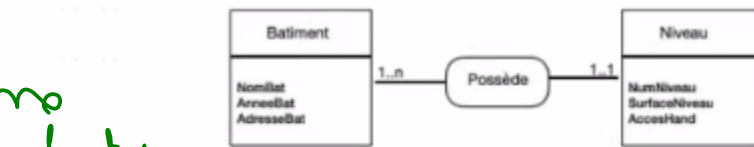
\* Dans le cahier des charges :

NomBat -> AnneeBat, AdresseBat  
 NomBat, NumNiveau -> SurfaceNiveau, AccesHand

\* Dans le schéma E/A proposé

les DF ne correspondent pas

et niveaux plus haut.

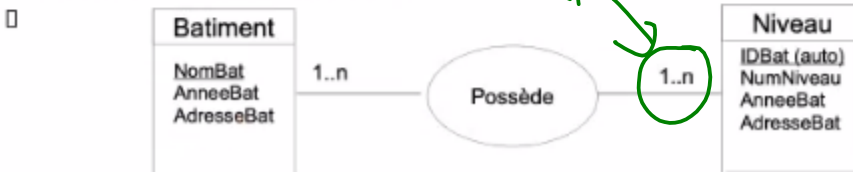


Batiment(NomBat, AnneeBat, AdresseBat)  
 Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)

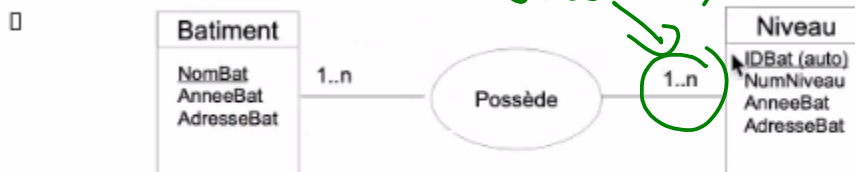
DF dans ce schéma relationnel :  
 NomBat -> AnneeBat, AdresseBat  
 NumNiveau -> NomBat, SurfaceNiveau, AccesHand

=> cahier des charges non respecté, car le numéro de niveau n'est pas unique dans l'établissement (il dépend du bâtiment).

Autre erreur « classique » de modélisation : inventer un identifiant unique pour « Niveau », par exemple en mode « auto-increment » :



Autre erreur « classique » de modélisation : inventer un identifiant unique pour « Niveau », par exemple en mode « auto-increment » :



Ne règle pas le problème, on a toujours perdu la DF :

NomBat, NumNiveau -> SurfaceNiveau, AccesHand

Ne marche pas si IDBat est auto-incrémenté dans Niveau sans lien avec Bâtiment

\* Mauvaise idée également  
de créer un identifiant par  
concaténation de nom de bâtiment  
et de numéro de niveau

LL → niveau A101  
                    ↑          ↑  
                  bâtiment  numéro  
                                niveau

mais il sera difficile d'accéder  
à ces informations par des  
requêtes SQL

3- Essayez de proposer intuitivement un bon schéma relationnel pour représenter les bâtiments et les étages.

Bâtiment ( NomBat : PK int,  
AnnéeBat : not null int,  
AdresseBat : not null text )

Niveau ( NumNiveau : not null int,  
NomBat : not null unique int FK references Bâtiment(NomBat) )

SurfaceNiveau : not null int  
AccessHand : not null boolean  
PrimaryKey (NumNiveau, NomBat) ,

PK: Primary Key ( def primaire )  
FK: Foreign Key ( def étrangère )



# Correction :

3- Essayez de proposer intuitivement un bon schéma relationnel pour représenter les bâtiments les étages.

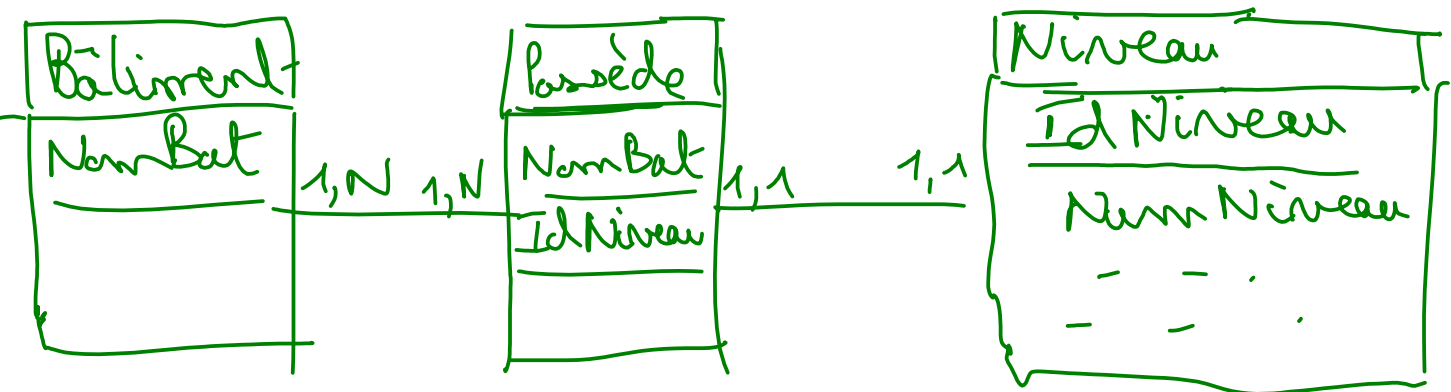
Si on modifie la relation « Niveau » pour intégrer la DF manquante, on obtient :

Batiment(NomBat, AnnéeBat, AdresseBat)  
Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)

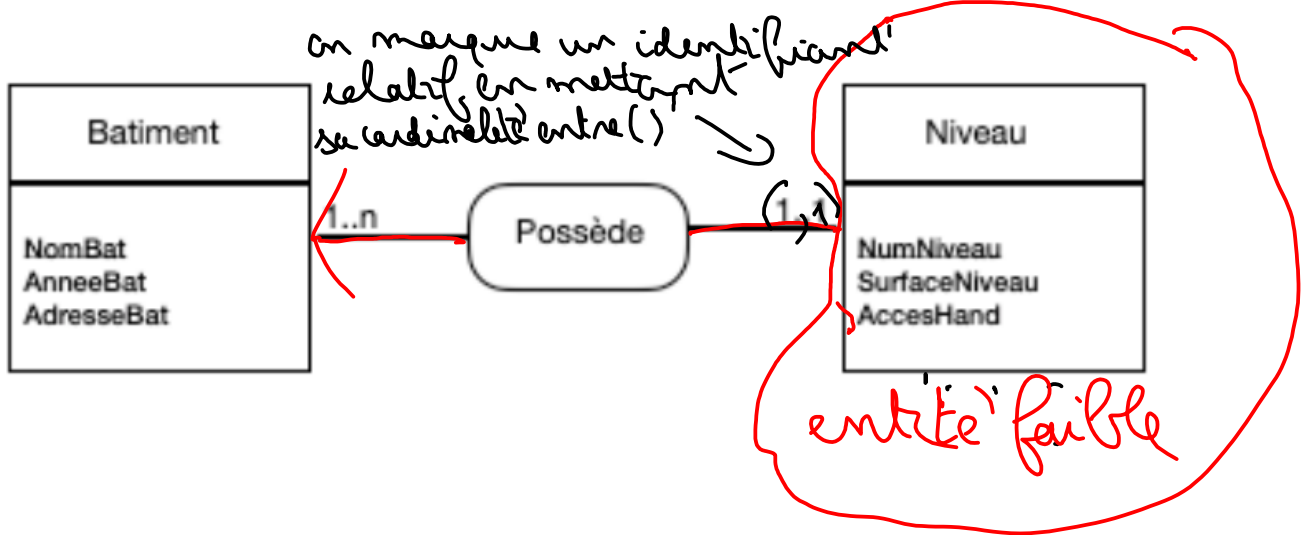
Cette fois, les spécifications sont bien respectées. Ce type de schéma relationnel correspond typiquement à la notion « d'entité faible » du langage E/A.

- On a intégré la clef étrangère # NomBat dans la clef primaire
- les dépendances fonctionnelles du cahier des charges sont bien respectées.

\* Une autre façon de faire



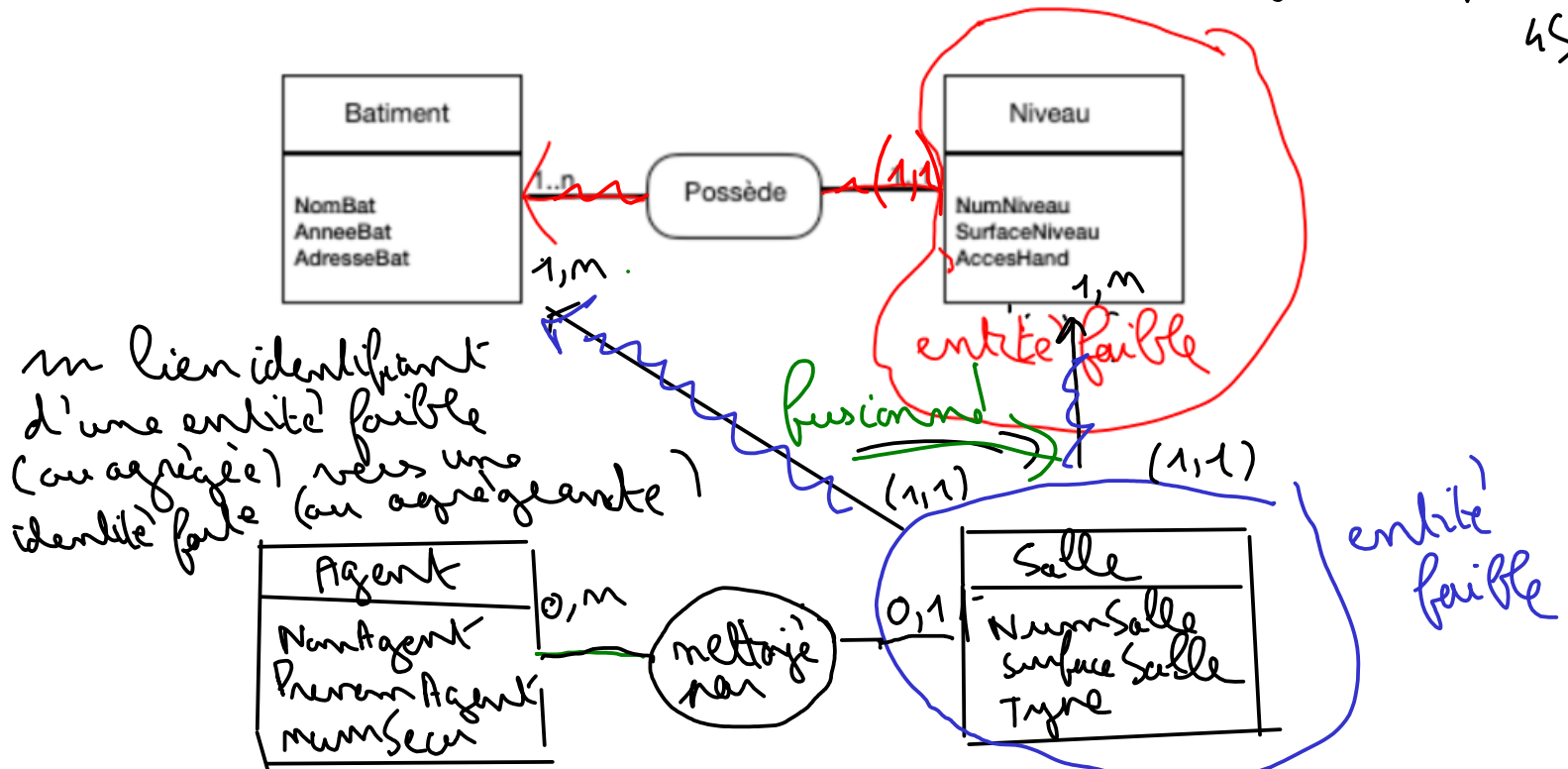
4- En vous appuyant sur la notion « **d'entité faible** », dressez un schéma E/A réalisant une modélisation correcte du problème.



5- Compléter le schéma pour intégrer les éléments suivants :

- Chaque salle de l'université est répertoriée ; elle possède un numéro unique au sein de l'étage, une surface et une type (Bureau, TD, TP, Cours, stockage etc...).
- Les agents d'entretien ont un nom, un prénom et un numéro de sécurité sociale qui est unique
- Chaque agent est affecté à un ensemble de salles de l'université ; chaque salle ne peut être associée qu'à un seul agent au maximum.

Entités faibles / Identifiant relatif / lien identifiant  
 => Audibert ps 44, 45



## Correction de 4):

l'identifiant local

4- En vous appuyant sur la notion « d'entité faible », dressez un schéma E/A réalisant une modélisation correcte du problème.

Une entité faible est une entité qui ne possède qu'une partie de son identifiant : celui-ci n'est pas « global » à l'application, mais sert à déterminer l'entité faible lorsqu'on connaît l'entité forte correspondante.

□



On dit que NumNiveau est l'identifiant « local » de Niveau ; un bâtiment peut bien avoir plusieurs niveaux dans les données, mais chaque niveau est attaché à un unique bâtiment qui partage son identifiant avec lui. La traduction en relationnel est bien :

Batiment(NomBat, AnneeBat, AdresseBat)

Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)

en JMerise, les entités faibles apparaissent avec la cardinalité entre chevrons <1,1>

est ce qu'on peut dire que "lorsque la clé primaire est composée d'au moins d'une clé étrangère , on est dans une entité faible?"

↳ si dans une clef primaire on a une clef étrangère probablement

. Si dans une clef primaire on n'a que des clefs étrangères c'est une association

- Valeur NULL :
  - NULL n'est pas comparable
  - Si une clef primaire est composite, le mot NULL porte sur chaque attribut.

PK = UNIQUE NOT NULL,  
le NOT NULL porte sur chaque attribut

6 – Traduisez le résultat obtenu en relationnel ; vérifiez bien, pour chaque DF qui pourrait être issue du cahier des charges, que sa partie gauche est bien une clé dans sa relation, et donc que le résultat est en FNBC.

Batiment ( NumBat : PK int,  
AnnéeBat : not null int,  
AdresseBat : not null text )

Niveau ( NumNiveau : not null int,  
NumBat : not null unique int FK references Batiment(NumBat)  

---

SurfaceNiveau : not null int  
AccessHand : not null boolean  
Primary Key (NumNiveau, NumBat) )

Salle ( NumSalle : not null int,  
SurfaceSalle : not null int,  
Type : not null text )

NumBat : unique not null int FK references Batiment(NumBat)  
NumNiveau : unique not null int FK references Niveau(NumNiveau)  
Primary Key (NumSalle, NumNiveau, NumBat)  
NumAgent : unique not null int FK references Agent(NumAgent)

↓ Fourn:

Foreign Key (NomBat, NumNiveau) References

Niveau (NomBat, NumNiveau);

Agent- (nomAgent: not null text,  
prenomAgent: not null text,  
numAgent- P K text)

— def étranger  
— def primaire

7- L'application évolue, et on souhaite intégrer la spécification suivante :

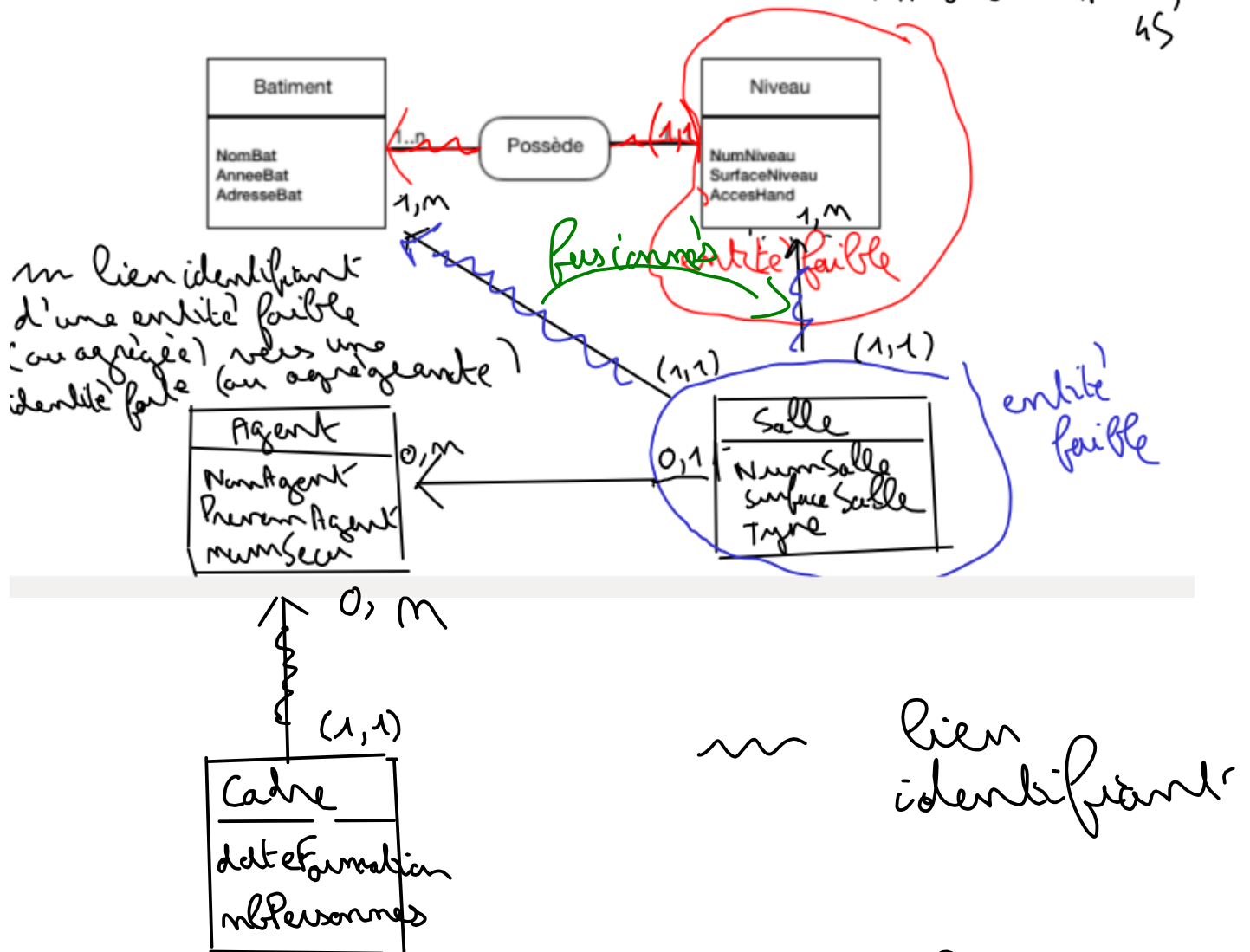
- Certains agents d'entretien ont un rôle particulier d'encadrement ; dans leur cas, on souhaite pouvoir gérer en plus leur date de dernière formation ainsi que le nombre de personnes encadrées.

En vous appuyant sur la notion de « **spécialisation** », proposer une évolution du diagramme E/A et répercuter le changement sur le modèle relationnel.

??



Entités faibles / Identifiant relatif / lien identifiant  
 ⇒ Audibert ns 44, 45

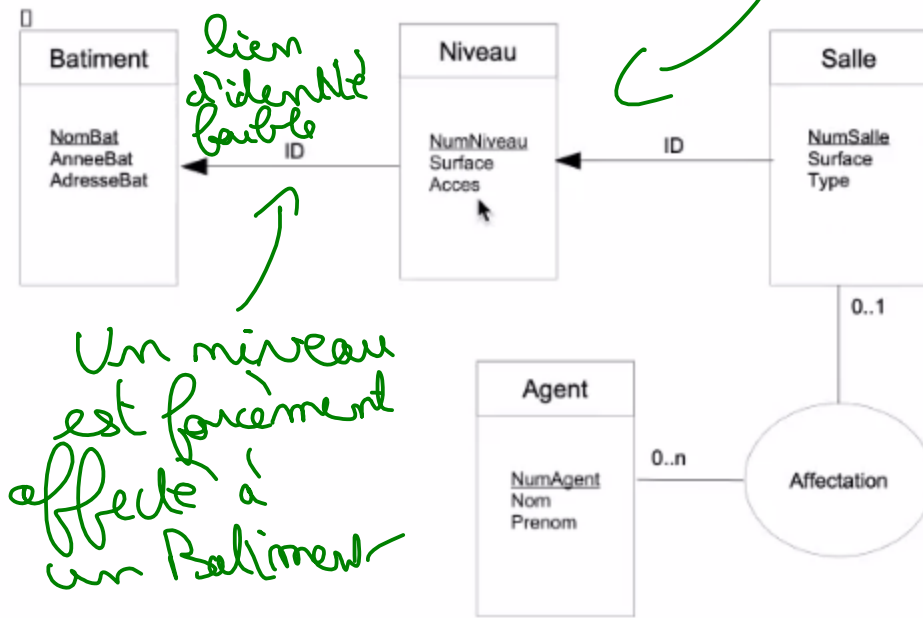


Dans le schéma relationnel  
 on rajoute une table.

Cadre ( dateFormation : not null date,  
 nbPersonnes : not null int,  
 numSecur : not null text FK references  
 Agent (numSecur),  
 PrimaryKey (numSecur) ).



## Correction de S)



Une salle est affectée à un niveau

Un niveau est forcément affecté à un Batiment

- Les cardinalités min sont fixées dans le cahier des charges
- Si on est strict, on exclut les valeurs NULL dans le modèle relationnel donc on va exclure les cardinalités min à 0.
- Dans la vraie vie, on accepte les valeurs NULL et les cardinalités min à 0

Sol 1

Batiment(NomBat, AnneeBat, AdresseBat)  
Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)  
Salle(NumSalle, #NumNiveau, #NomBat, Surface, Type)  
Agent(NumAgent, Nom, Prenom)  
Affectation(#NumAgent, #NumSalle, #NumNiveau, #NomBat)

Où encore, on peut aussi traduire avec le schéma suivant :

Sol 2

Batiment(NomBat, AnneeBat, AdresseBat)  
Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)  
Salle(NumSalle, #NumNiveau, #NomBat, #NumAgent, Surface, Type)  
Agent(NumAgent, Nom, Prenom)

#NumAgent non souligné car une affectation est entièrement caractérisée par une salle (on est sur une table traduisant une association)

$1:n$  association  $1:1$  identifiant  
relation fonctionnelle

\* L'avantage de la solution 1 est qu'elle permet de ne pas affecter d'agent au nettoyage d'une salle sans insérer de valeurs NULL dans la table SALLÉ

Batiment(NomBat, AnneeBat, AdresseBat)  
Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)  
Salle(NumSalle, #NumNiveau, #NomBat, Surface, Type)  
Agent(NumAgent, Nom, Prenom)  
Affectation(#NumAgent, #NumSalle, #NumNiveau, #NomBat)

Ou encore, on peut aussi traduire avec le schéma suivant :

Batiment(NomBat, AnneeBat, AdresseBat)  
Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)  
Salle(NumSalle, #NumNiveau, #NomBat, #NumAgent, Surface, Type)  
Agent(NumAgent, Nom, Prenom)

Remarque : il est important de mieux spécifier certaines clés étrangères, pour lever les ambiguïtés, et en particulier :

Salle[NumNiveau, NomBat] INCLU DANS Niveau[NumNiveau, NomBat]

\* Les deux schémas relationnels sont justes et traduisent le même diagramme E/A

```
CREATE TABLE Affectation(  
  NumSalle int,  
  NumNiveau int,  
  NomBat text,  
  NumAgent int,  
  PRIMARY KEY (NumSalle, NumNiveau, NomBat),  
  FOREIGN KEY (NumAgent) REFERENCES Agent(NumAgent),  
  FOREIGN KEY (NumSalle, NumNiveau, NomBat) REFERENCES Salle(NumSalle, NumNiveau, NomBat)  
)
```

clé étrangère de taille 3  
⇒ à bien distinguer de  
trois clés étrangères  
de taille 1.

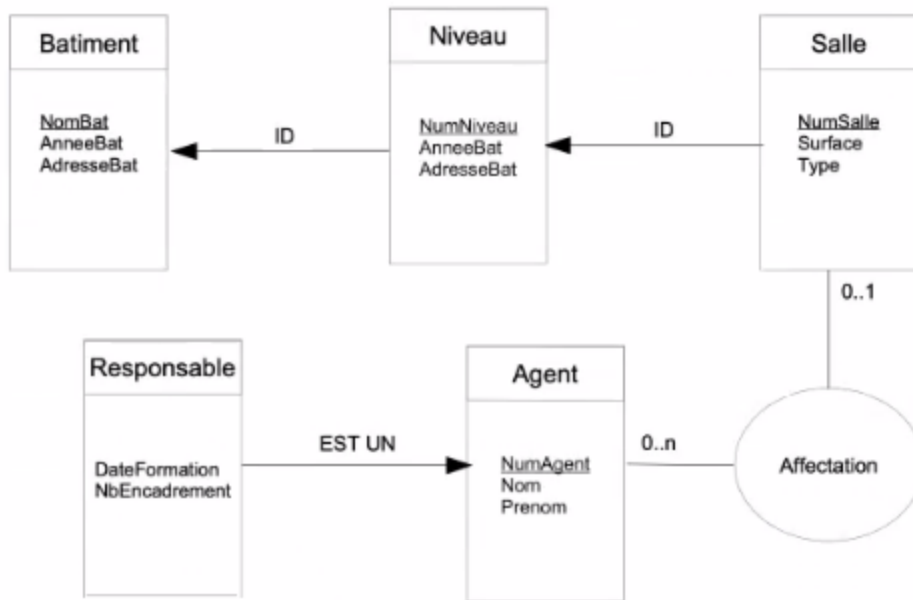
existence  
d'une  
salle

## Correction de la question 7 :

7- L'application évolue, et on souhaite intégrer la spécification suivante :

- Certains agents d'entretien ont un rôle particulier d'encadrement ; dans leur cas, on souhaite pouvoir gérer en plus leur date de dernière formation ainsi que le nombre de personnes encadrées.

En vous appuyant sur la notion de « **spécialisation** », proposer une évolution du diagramme E/A et répercuter le changement sur le modèle relationnel.



L'entité « responsable » est une entité de spécialisation. Elle n'a pas d'identifiant propre, car chaque objet de ce type dans les données est déjà identifié par son numéro d'agent.

- Pas d'identifiant propre pour Responsable, c'est une sous-entité, elle s'apparente à une relation d'héritage en POO.
- Une entité de spécialisation n'est pas une entité faible. (elle n'a pas d'identifiant propre)

elle est identifiée par l'entité dont elle hérite)

- Un responsable possède un numéro d'agent - mais dans le diagramme E/A on ne répète pas les attributs

L'entité « responsable » est une entité de spécialisation. Elle n'a pas d'identifiant propre, car chaque objet de ce type dans les données est déjà identifié par son numéro d'agent.

La traduction de cette entité dans le modèle relationnel est :

Responsable (#NumAgent, DateFormation, NbEncadrement)

• Une entité de spécialisation n'a pas d'identifiant  $\Rightarrow$  lorsqu'on traduit en modèle relationnel, sa clé primaire est une clé étrangère : la clé primaire de l'entité dont elle hérite.