**TP4\_conception\_de\_Batiments**

Exercice : variations sur le thème entité faible et spécialisation

L’université souhaite gérer ses bâtiments et leur entretien.

On donne les **spécifications** suivantes : (**cahier des charges**)

- 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.

On donne la liste des **attributs** issus de ces spécifications :

NomBat, AnneeBat, AdresseBat, NumNiveau, SurfaceNiveau, AccesHand.

**1-** Dressez l’inventaire des **dépendances fonctionnelles (DF)** entre les attributs.

DF1 NomBat AnneeBat, AdresseBat

DF2 NomBat ,NumNiveau SurfaceNiveau, AccesHand

**2-** Voici un **diagramme entité/association (E/A)** qui propose un extrait de modélisation en ce qui concerne les bâtiments et les étages.

Remarque : Jamais deux fois le même attribut dans un diagramme E/A ! mettre des noms différents !

Les **cardinalités** sur les pattes indiquent qu’un bâtiment peut avoir entre 1 et plusieurs niveaux

Mais qu’un niveau n’appartient qu’à un seul bâtiment.

Bâtiment et Niveau sont des **entités**, Possède est une **association**, les autres **attributs**.

Batiment

NomBat

AnneeBat

AdresseBat

Niveau

NumNiveau

SurfaceNiveau

AccesHand

Possède

1..n

1..1

Traduisez ce diagramme en **schéma relationnel**, puis critiquez le résultat en vous appuyant sur les DF trouvées.

Remarque : Chaque entité devient toujours une relation.

On doit retrouver nos DF.

Batiment(NomBat, AnneeBat, AdresseBat)

Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)

Remarque : Le dièse # traduit l’association Possède en déclarant NomBat comme **clé étrangère (FK)**

En partant de ce schéma, retrouvons les DF : Problème !

NomBat AnneeBat, AdresseBat

NumNiveau NomBat, SurfaceNiveau, AccesHand

Le cahier des charges n’est pas respecté : le numéro de niveau n’est pas unique, il dépend du bâtiment.

Remarque : Une association peut toujours devenir une relation, à ce moment dans « Possede » il y des couples qui associent. Ou bien, on en fait une simple clé étrangère.  
C’est possible de s’en contenter d’en faire un attribut (une référence) de « Niveau ».

C’est le cas lorsque les cardinalité max sont 1.n (max d’un côté : 1 et n : de l’autre).

Pour les min, il vaut mieux écrire 0 plutôt que 1 pour garder de la souplesse dans la création.

**Autre erreur classique de modélisation :** inventer un identifiant unique pour Niveau en mode auto-incrément.

Ajouter l’attribut idBat dans l’entité Niveau ne règle pas le pb : on aura toujours perdu la DF2 du départ !

**Autre mauvaise idée :** créer un identifiant par concaténation… Comme par exemple la salle A103 pour la salle 03, à l’étage 1 du bâtiment A. Il vaut mieux séparer les infos pour simplifier leur accès par requêtes SQL.

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

L’idée est de faire une **clé primaire (PK) composite**.

Bon schéma relationnel qui respecte le cahier des charges et les DF :

Batiment(NomBat, AnneeBat, AdresseBat)

Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)

NomBat est PK pour Batiment et fait partie de la PK composite de Niveau et c’est aussi une FK pour Niveau.

Remarque : Ce type schéma relationnel correspond typiquement à la notion d’**entité faible** du langage E/A.

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.

Elle a besoin d’une référence à une entité forte pour exister.

Si, dans une PK, il y a une FK, c’est probablement une entité faible.

Si, dans une PK, il n’y a que des FK, ce sont probablement des associations.

Remarque : PK implique UNIQUE et NOT NULL

FK peut être NULL

Si NOT NULL porte sur une clé composite, chaque attribut est NOT NULL.

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

Batiment

NomBat

AnneeBat

AdresseBat

Niveau

SurfaceNiveau

AccesHand

**ID**

**ou <1,1>**

Entité faible

un bâtiment peut avoir plusieurs niveaux mais chaque niveau est attaché à un unique bâtiment qui partage son identifiant avec lui.

Identifiant local

**5-** Complétez le diagramme 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 un 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.

Batiment

NomBat

AnneeBat

AdresseBat

Niveau

NumNiveau

SurfaceNiveau

AccesHand

**ID**

Salle

NumSalle

SurfaceSalle

Type

**ID**

Agent

NumAgent

Nom

Prenom

affectation

0..n

0..1

Liens d’identité faible :

1 niveau est forcément affecté à 1 bâtiment

1 salle est forcément affecté à 1 niveau

Remarque : Une entité faible peut pointer sur une entité faible (juste un peu plus forte…).

Elle pourrait pointer sur 2 entités, mais c’est déconseillé.

Agent n’est pas une entité faible de Salle car il n’a besoin que de ses attributs pour être identifier.

**6-** Traduisez le résultat obtenu en schéma 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(NomBat, AnneeBat, AdresseBat)

Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)

Salle(NumSalle, #NumNiveau, #NomBat, SurfaceSalle, Type)

Agent(NumAgent, Nom, Prenom)

NomBat AnneeBat, AdresseBat

NomBat ,NumNiveau SurfaceNiveau, AccesHand

NumSalle, NumNiveau, NomBat SurfaceSalle, Type

NumAgent, Nom, Prenom

Remarque : Chaque entité forte ou faible fait une relation.

La clé primaire d’une identité faible a besoin de la PK de l’entité forte vers qui elle pointe.

**Autre solution** : faire de l’association « affectation » une relation.

Batiment(NomBat, AnneeBat, AdresseBat)

Niveau(NumNiveau, #NomBat, SurfaceNiveau, AccesHand)

Salle(NumSalle, #NumNiveau, #NomBat, SurfaceSalle, Type)

Agent(NumAgent, Nom, Prenom)

Affectation(#NumAgent, #NumSalle, #NumNiveau, #NomBat)

Remarque : NumAgent non souligné ici, car d’après le cahier des charges une salle est nettoyée par au plus un agent. Affectation entièrement caractérisée par la salle…

Avantage de la deuxième solution :

elle permet de ne pas affecter d’agent au nettoyage d’une salle sans insérer de valeurs NULL dans la table Salle.

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]

**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 »**, proposez une évolution du diagramme E/A et répercutez le changement sur le modèle relationnel.

Batiment

NomBat

AnneeBat

AdresseBat

Niveau

NumNiveau

SurfaceNiveau

AccesHand

**ID**

Salle

NumSalle

SurfaceSalle

Type

**ID**

0..1

0..n

Agent

NumAgent

Nom

Prenom

affectation

Responsable

DateFormation

NbEncadr

**EST UN**

On va donner des rôles différents à des sous types d’entités.

Certains agents ont des particularités mais pas tous !

La spécialisation n’est pas une entité faible, elle est encore moins forte... elle n’a pas d’identifiant propre.

On ne met pas d’identifiant local pour une spécialisation dans le langage E/A.

La PK d’une spécialisation est toujours celle de l’entité dont elle hérite ; elle est clé native dans l’entité principale et clé étrangère dans la spécialisation.

Attention ici pour l’écriture en SQL…

Bien faire la différence entre une FK de taille 3 (ici) et 3 FK de taille1.

///// Script : attention, le script n'est pas rigoureux dans un outil donné, il doit être adapté en ce qui concerne les types

CREATE TABLE Batiments(

NomBat : VARCHAR,

AnneeBat : DATE,

AdresseBat : VARCHAR,

PRIMARY KEY(Nombat)

)

CREATE TABLE Niveaux(

NomBat : VARCHAR,

NumNiveau : INT,

SurfaceNiveau : INT,

AccesH : BOOL,

PRIMARY KEY(NomBat, NumNiveau),

FOREIGN KEY (NomBat) REFERENCES Batiments(NomBat)

)

CREATE TABLE Salles(

NomBat : VARCHAR,

NumNiveau : INT,

NumSalle : INT,

SurfaceSalle : INT,

Type : VARCHAR,

PRIMARY KEY(NomBat, NumNiveau, NumSalle),

FOREIGN KEY (NomBat, NumNiveau) REFERENCES Salles(NomBat, NumNiveau)

)

CREATE TABLE Agents(

NumAgentt : INT,

Nom : VARCHAR,

Prenom : VARCHAR,

PRIMARY KEY(NumAgent),

)

CREATE TABLE Affectation(

NumSalle INT,

NumNiveau INT,

NomBat VARCHAR,

NumAgent INT,

PRIMARY KEY (NumSalle, NumNiveau, NomBat),

FOREIGN KEY (NumSalle, NumNiveau, NomBat) REFERENCES Salles(NumSalle, NumNiveau, NomBat) ,

FOREIGN KEY (NumAgent) REFERENCES Agents(NumAgent)

)

CREATE TABLE Responsables(

NumAgent : INT,

DateFormation : DATE,

NbEncadrements : INT,

PRIMARY KEY(NumAgent),

FOREIGN KEY (NumAgent) REFERENCES Agents(NumAgent)

)