

Du diagramme de classe UML à la création des bases de données



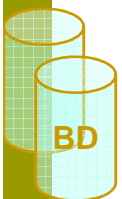
Quelques notions sur le **Langage de
Définition de **D**onnées**

Présentation

La conception du diagramme de classe UML est une étape importante vers la création des BDD, mais il faut ensuite implémenter ce diagramme dans un SGBDR.

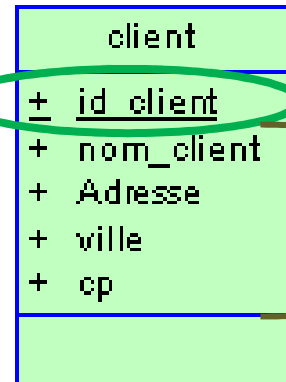
Il existe deux étapes pour arriver à la création des tables à partir du diagramme de classe:

1. Création du **Modèle Physique des Données (MPD)**;
2. Création du script SQL de génération des tables à l'aide du **Langage de Définition de Données (LDD)**;

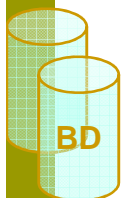
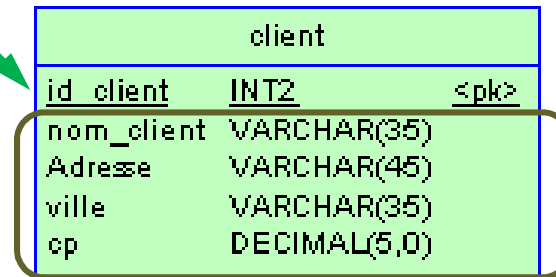


Une relation

UML



MPD

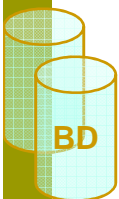


Une relation(2)

Définition de données

client		
<u>id_client</u>	<u>INT2</u>	<u><pk></u>
nom_client	VARCHAR(35)	
Adresse	VARCHAR(45)	
ville	VARCHAR(35)	
cp	DECIMAL(5,0)	

```
CREATE TABLE client
(
  id_client    int           primary key,
  nom_client   varchar(35) ,
  adresse      varchar(45),
  ville        varchar(35),
  CP           decimal(5,0)
);
```

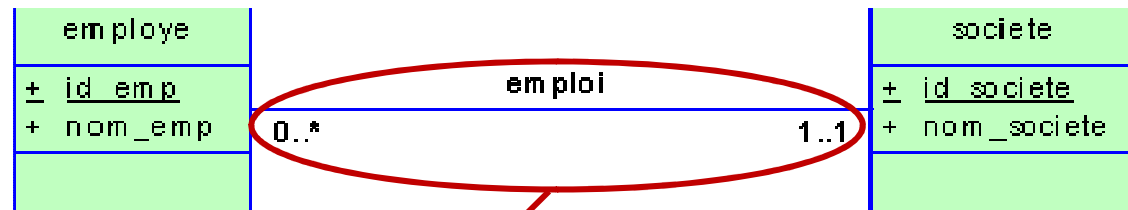


Association binaire (1)

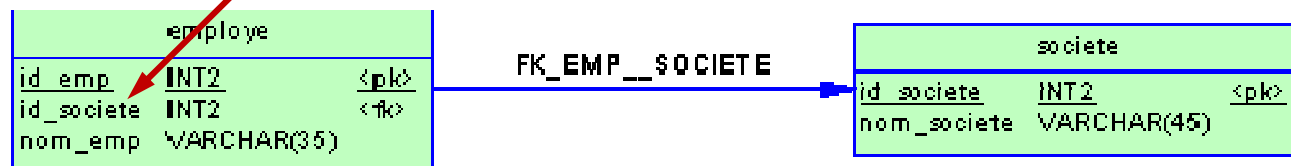
cardinalités (X..1) - (X..*) [X=0 ou X=1]

Un des deux maximum = 1

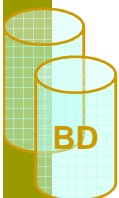
UML



MPD

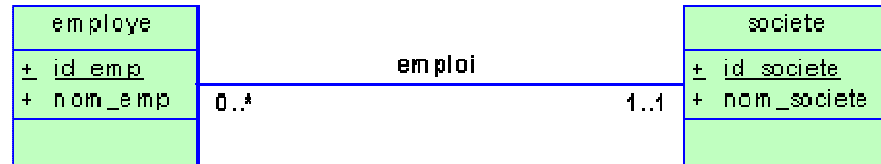


id_societe de la table employe est une clé étrangère <fk> qui fait référence à la colonne id_societe de la table societe



Notion de clé étrangère

UML



MPD



La clé étrangère fait référence au champs **id_societe** de la table **SOCIETE**

Que doit vérifier la règle d'intégrité référentiel ?

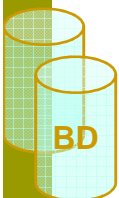
EMPLOYE		
Id_emp	id_societe	nom_emp
100	10	Bidule
120	10	Machin
130	20	Truc
140	20	Carmi
150	20	Dupond
160	50	Durant

Est-ce que la société N°10 existe

SOCIETE	
Id_societe	Nom_societe
10	InfoWeb
20	geoconcept
30	infograph

Erreur ! Pas de société N°50

Elle vérifie la cohérence entre les valeurs de clé étrangère et celle de la clé primaire de la table référencée



Association binaire (2)

cardinalités (X..1) - (X..*) [X=0 ou X=1]

Un des deux maximum = 1

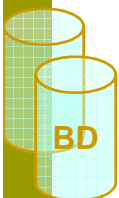
Définition de données

```
CREATE TABLE SOCIETE
```

```
(  
  id_societe    int           primary key,  
  nom_societe   varchar(45) null  
);
```

```
CREATE TABLE EMPLOYE
```

```
(  
  id_emp        int           primary key,  
  id_societe    int           not null,  
  nom_emp       varchar(35)   default null,  
  constraint FK_EMP__SOCIETE foreign key (id_societe)  
                                references SOCIETE (id_societe)  
);
```

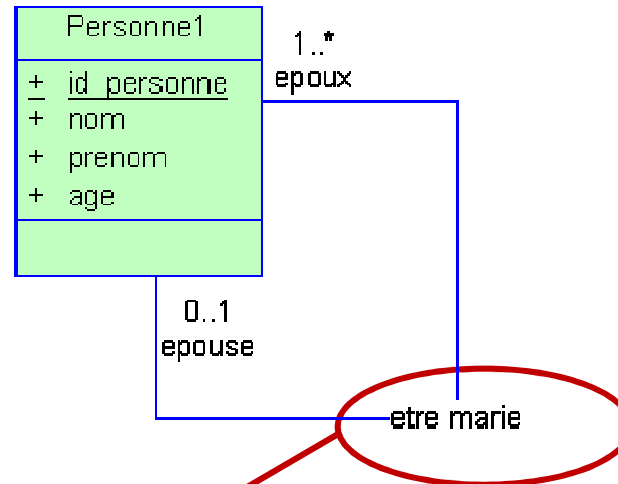


Association binaire reflexive (1)

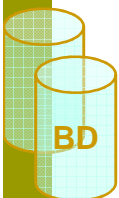
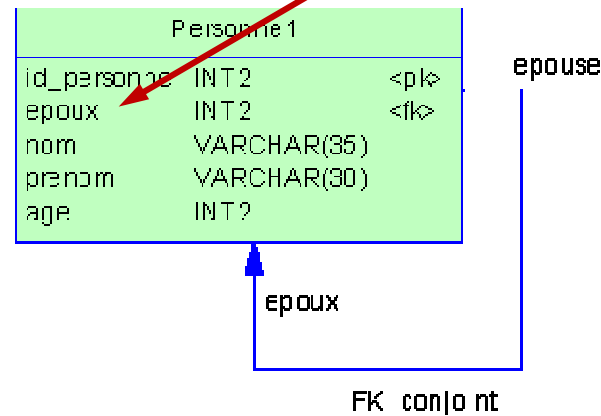
cardinalités (X..1) - (X..*) [X=0 ou X=1]

Un des deux maximum = 1

UML



MPD



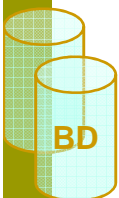
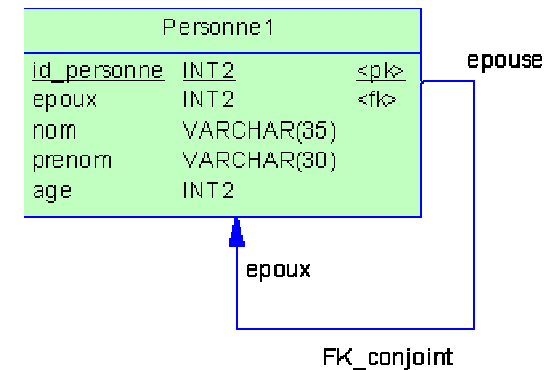
Association binaire reflexive (2)

cardinalités (X..1) - (X..*) [X=0 ou X=1]

Un des deux maximum = 1

Définition de données

```
CREATE TABLE personne1
(
  id_personne  int      primary key,
  epoux        int,
  nom          varchar(35) default null,
  prenom       varchar(35) default null,
  age          int not null,
  constraint FK_personne foreign key (epoux)
                    references personne1(id_personne)
);
```

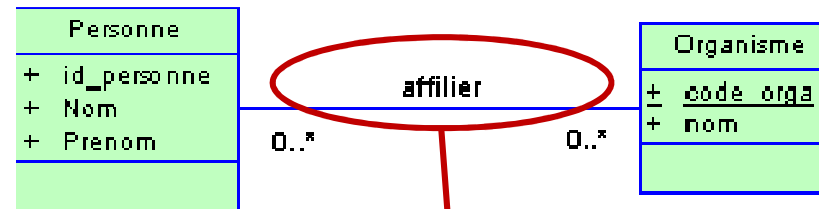


Association binaire (1)

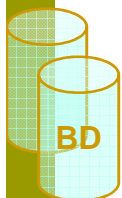
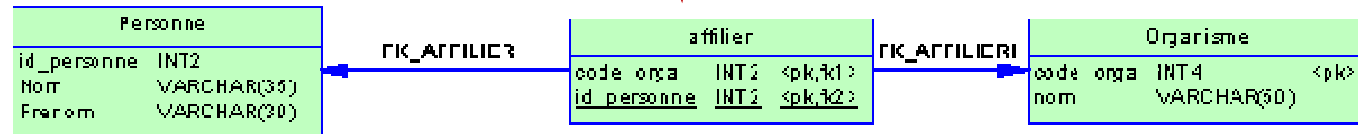
cardinalités (X..*) - (X..*) [X=0 ou X=1]

Les deux maximum > 1

UML



MPD



Association binaire (2)

cardinalités (X..*) - (X..*) [X=0 ou X=1]

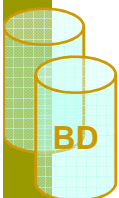
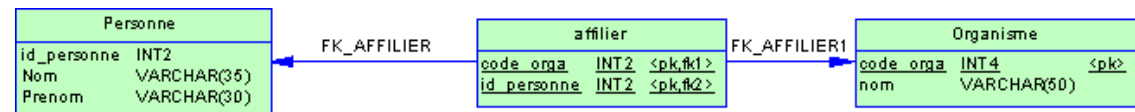
Les deux maximum > 1

Définition de données

```
create table personne
( id_personne int primary key,
  nom          varchar(35),
  prenom       varchar(35) );
```

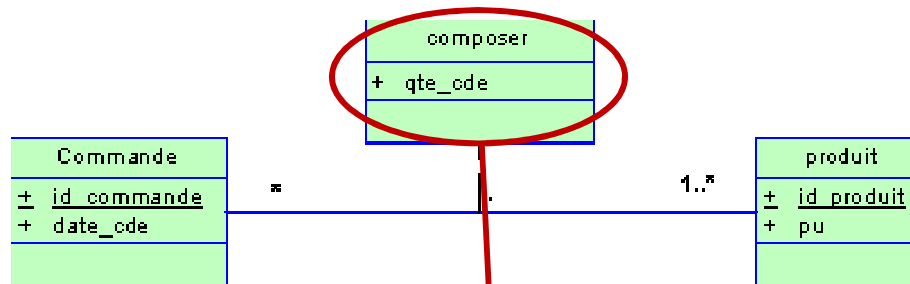
```
create table organisme
( code_organisme int primary key,
  nom            varchar(50) null );
```

```
create table affilier
( id_personne int ,
  code_organisme int ,
  constraint PK_affilier primary key (id_personne , code_organisme ),
  constraint FK_affilier foreign key (id_personne)
                                references personne(id_personne),
  constraint FK_affilier1 foreign key (code_organisme)
                                references organisme(code_organisme)
);
```

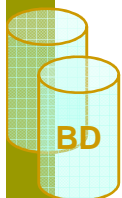
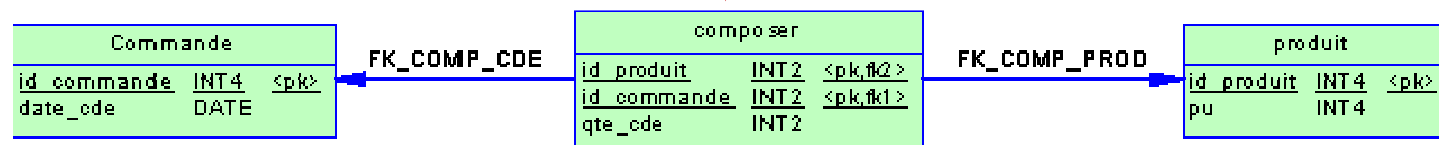


Classe d'association (1)

UML →



MPD →



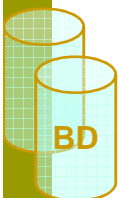
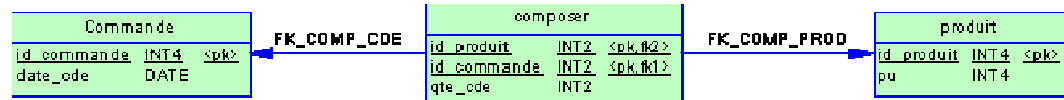
Classe d'association (1)

Définition de données

```
create table commande
( id_commande int primary key,
  date_cde    date);
```

```
create table produit
( id_produit int      primary key,
  pu         int      null );
```

```
create table composer
( id_commande int      not null,
  id_produit  int      not null,
  qte_cde     int      default 1,
  constraint PK_COMP primary key ( id_commande,id_produit ),
  constraint FK_COMP foreign key (id_commande) references COMMANDE(id_commande),
  constraint FK_COMP_P foreign key (id_produit) references PRODUIT (id_produit)
);
```

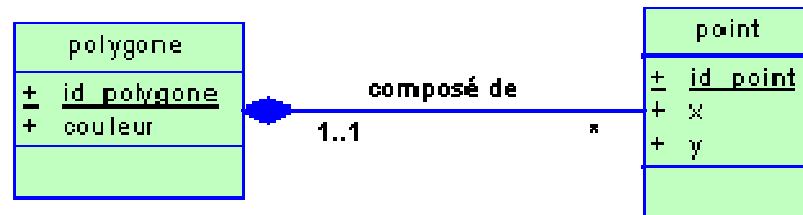


Agrégation et composition(1)

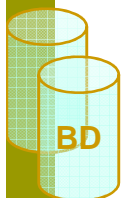
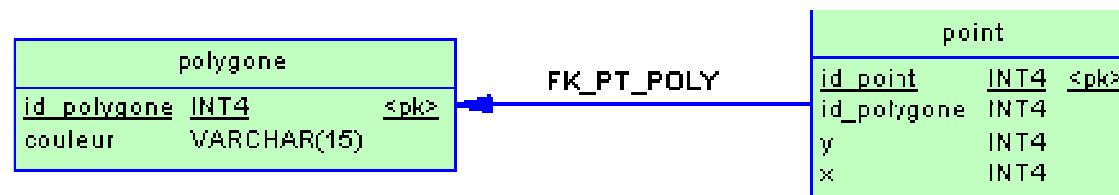
Elles se traitent comme les associations binaires, la dépendance (agrégation ou composition) indiquera la suppression en cascade ou non

Exemple de la composition

UML

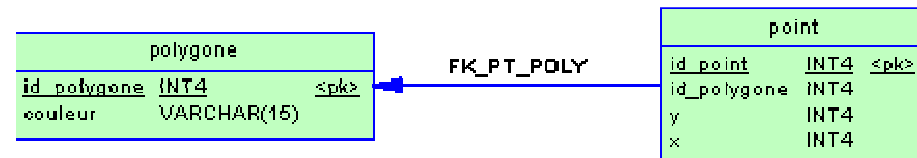


MPD



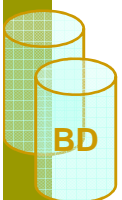
Agrégation et composition(2)

Définition de données



```
create table polygone
( id_polygone int primary key,
  couleur      varchar(15));
```

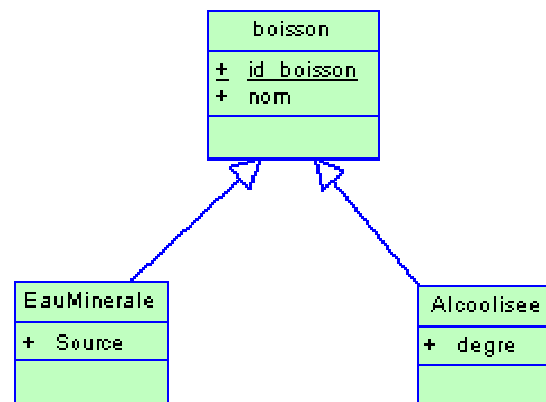
```
create table point
( id_point      int primary key,
  id_polygone   int not null,
  x             int ,
  y             int ,
  constraint FK_poly foreign key (id_polygone )
               references polygone(id_polygone ) on delete cascade
);
```



L'héritage(1)

Solution tout en un : elle consiste à mettre tous les attributs dans la même table

UML

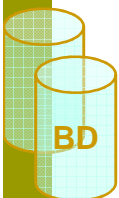


MPD



boisson		
<u>id_boisson</u>	INT4	<pk>
nom	INT4	
genre	CHAR(15)	
source	CHAR(25)	
degre	DECIMAL(3,1)	

**Genre peut prendre les valeurs :
Eau ou Alcoolisee**



Fin

