

# Concevoir une base de données

[sylvain.parise@ac-bordeaux.fr](mailto:sylvain.parise@ac-bordeaux.fr) - CC-BY-NC



# Croisières

Une agence de voyage propose des croisières en bateau. Chaque croisière a un nom [...] et passe par [...] [différentes] escales correspondant à des villes [...].

Pour gérer les réservations de ses clients, l'agence utilise une base de données.

Une croisière propose de une à dix escales. Chaque escale est spécifique à la croisière.

Une escale doit proposer des activités dont on ne proposera qu'une description sommaire ('Ballade de 3 km avec point de vue', 'Traverser d'un marché local', 'Visite de la basilique') et d'une durée approximative en heure. L'escale sera également décrite par sa durée en heure, la ville visitée, les villes à proximité avec le nombre de kilomètres pour s'y rendre.

L'agence désire mémoriser pour chaque ville, le pays, la monnaie et un descriptif sous forme de texte des formalités nécessaire pour accoster.

Lorsqu'un client réserve pour une croisière, il doit préciser les nombres de personnes adultes et enfants qui l'accompagnent. Bien entendu, il ne peut acheter qu'une seule réservation par croisière. Un client sera caractérisé par son nom.

Pour simplifier, on supposera que chaque croisière est unique, même si la même croisière est organisée périodiquement.



**Client**(idClient, nom, prenom, dateNaiss, pays)

**Réservations**(idReservation, #idClient, #nomCroisiere)

**Croisière**(nom, escale1, escale2, escale3, escale4, escale5, escale6, escale7, escale8, escale9, escale10, activitésEscale1, activitésEscale2... *pfff c'est long*)

**Ville**(nom, latitude, longitude, *eh non...*)

Et comment écrire la requête suivante :  
« Quelles sont les croisières qui passent par  
Bordeaux ? »

```
Select nom from Croisière where escale1 = 'Bordeaux' or escale2  
= 'Bordeaux' or escale3 = 'Bordeaux' or escale4 = 'Bordeaux' or  
escale5 = 'Bordeaux' or escale6 = 'Bordeaux' or escale7  
= 'Bordeaux' or escale8 = 'Bordeaux' or escale9 = 'Bordeaux' or  
escale10 = 'Bordeaux'
```

# Gestion de CD

*Bob, qui dispose d'une très grande collection de CDs rangés sur plusieurs étagères numérotées, veut mettre en place une base de données.*

Pour simplifier, on supposera qu'il n'y a qu'un seul artiste par album ; cet artiste peut être une personne physique ou un groupe de personnes. On ne détaillera pas la composition du groupe. On ne gère pas des albums de compilations de titres de différents artistes.

Pour Bob, un album est décrit par son nom, son artiste principal, le nombre de morceaux, la durée totale en seconde de l'album, le genre principal de cet album et l'année de publication de celui-ci. La liste des titres de l'album avec leur durée en seconde respective intéresse aussi Bob pour faire des recherches.

Un album est principalement sous la forme d'un CD ou d'un DVD, mais il existe d'autres supports (fichiers numérique, vinyles, etc.).

Bob veut également enregistrer les styles musicaux où s'illustre l'artiste, le pays d'origine et son âge approximatif et son éventuelle date de décès.

Enfin, Bob est très ordonné. Il range systématiquement les albums sur la même étagère. Pour informatiser cela, il a numéroté chacune d'elle. Il mémorise également une description de l'emplacement ('étagère du haut', 'sous la TV', etc.). Il convient de mémoriser sur quelle étagère il peut retrouver son album. Attention, toutes les étagères ne peuvent pas contenir n'importe quel support ; certaines peuvent accueillir des CD ou des DVD, mais pas des vinyles. Toutes les combinaisons sont possibles.



# Gestion de CD

Album(titreAlbum, nomArtiste, nbMorceau, durée, styleMusical, année, support, numEtagère)  
Album(titreAlbum, #nomArtiste, nbMorceau, durée, styleMusical, année, support, numEtagère)

Morceaux(titreMorceaux, durée, #titreAlbum)

Artiste(nomArtiste, stylesMusical, paysOrigine, age, dateDécès)



groupe de personnes. On ne détaillera pas la composition  
albums de compilations de titres de différents artistes.

Pour Bob, un album est décrit par son nom, son artiste principal, le nombre de morceaux,  
principal de cet album et l'année de  
publication de celui-ci. La liste des titres de l'album avec leur durée en seconde respective  
intéresse aussi Bob pour faire des recherches.

Un album est principalement sous la forme d'un CD ou d'un DVD, mais il existe d'autres  
supports (fichiers numérique, vinyles, etc.).

Bob veut également enregistrer les styles musicaux où s'illustre l'artiste, le pays d'origine  
et son âge approximatif et son éventuelle date de décès.

Enfin, Bob est très ordonné. Il range systématiquement les albums sur la même étagère.  
Pour informatiser cela, il a numéroté chacune d'elle. Il mémorise également une  
description de l'emplacement ('étagère du haut', 'sous la TV', etc.). Il convient de  
mémoriser sur quelle étagère il peut retrouver son album. Attention, toutes les étagères  
ne peuvent pas contenir n'importe quel support ; certaines peuvent accueillir des CD ou  
des DVD, mais pas des vinyles. Toutes les combinaisons sont possibles.

**01**

## Le problème Prince

Dans les années 90 en raison d'un désaccord avec sa maison d'édition, Prince interdit qu'on l'appelle Prince. Il préfère LoveSymbol...



Artiste(nomArtiste, stylesMusical, paysOrigine, age, dateDécès)

**02**

## La toune cachée

Lorsque Bob saisit un CD dans sa base de données, il prend la jaquette et recopie les informations. Il calcule la durée totale et compte le nombre de morceaux pour mettre à jour sa table.

A l'écoute, il s'aperçoit qu'il y a un morceau bonus, caché à la fin du CD...

Album(titreAlbum, #nomArtiste, nbMorceau, durée, styleMusical, année, support, numEtagère)

**03**

## L'album IV de Led Zep'

Lorsqu'ils commencèrent à être un peu connus, les Led Zep essayèrent d'imposer leur vision d'artiste à leur maison de production. Le 4<sup>e</sup> album du groupe n'aura pas de nom...

**04**

## Les styles de l'artiste ?

Le champs de la table Artiste contient plusieurs valeurs. C'est en effet, styleS au pluriel. Si c'est très satisfaisant pour mémoriser, c'est plus difficile pour effectuer une recherche.

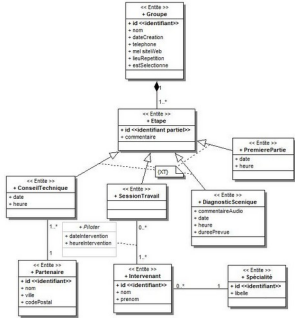



- Comment passer de la description d'un contexte à un script SQL ?

On veut une BDD pour stocker les règlements des factures (montants, dates, ...). On voudra différencier les règlements en cheque (on stockera le nom de la banque et le numéro du chèque) et ceux en carte bancaire (on stockera l'empreinte du numéro de carte et la date d'expiration)....

```
create table reglement (  
  numReglement number,  
  dateReglement date,  
  montant number,  
  numFacture number,  
  constraint PK_ REGLEMENT primary key (numReglement),  
  constraint FK_ REGLEMENT_FACTURE foreign key  
  (numFacture) references facture(num)  
);
```

# Modélisation



 - Comment passer de la description d'un contexte à un script SQL ?

Adaptation héritée



On veut une BDD pour stocker les règlements des factures (montants, dates, ...). On voudra différencier les règlements en cheque (on stockera le nom de la banque et le numéro du chèque) et ceux en carte bancaire (on stockera l’empreinte du numéro de carte et la date d’expiration)....

```
create table reglement (
    numReglement number,
    dateReglement date,
    montant number,
    numFacture number,
    constraint PK_ REGLEMENT primary key (numReglement),
    constraint FK_ REGLEMENT_FACTURE foreign key
    (numFacture) references facture(num)
);
```





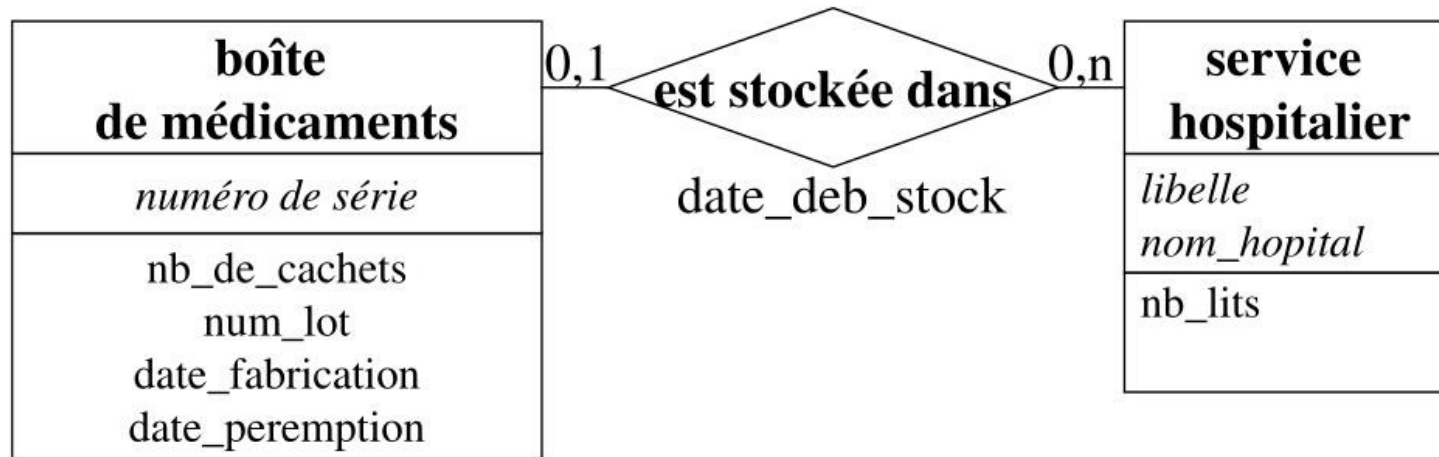


- Un peu d'histoire

Non, ce n'est pas moi.



# Modèle de Chen



Apparu dans les années 1970, il jette les bases de l'abstraction nécessaire à la conception des bases de données relationnelles.



# MERISE

Plus qu'un schéma, René Colletti, Arnold Rochfeld et Hubert Tardieu proposent une méthodologie pour automatiser un système d'information de manière systémique. Par exemple : le calcul des fiches de paie des fonctionnaires d'état.

## Ce n'est pas un acronyme

Il faut y voir une analogie avec le merisier « qui ne peut porter de beaux fruits que si on lui greffe une branche de cerisier : ainsi en va-t-il des méthodes informatiques bien conçues, qui ne produisent de bons résultats que si la greffe sur l'organisation réussit »

## Courbe en soleil

La méthode propose de s'abstraire des contraintes physiques pour concevoir « à l'idéal » le SI.  
Puis la démarche redescend en s'adaptant aux contraintes physiques, tel le soleil couchant.

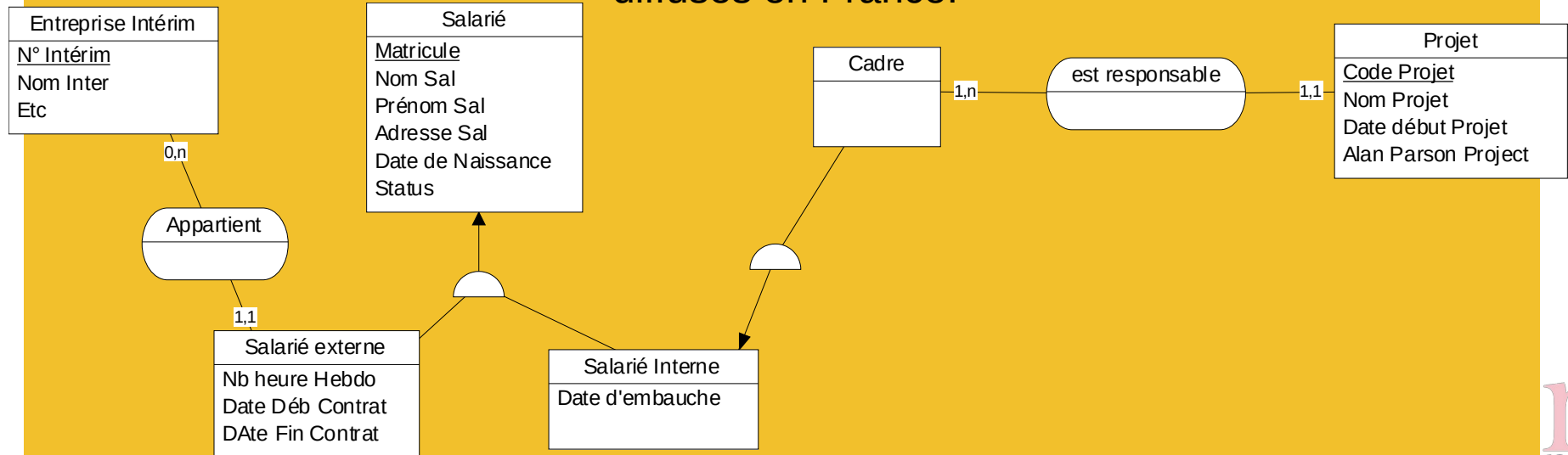
## Un outil graphique

La méthode propose une dizaine de schémas parmi lesquels on retrouve le modèle de Chen, appelé ici MCD pour Modèle Conceptuel des Données.



# MERISE

La dernière mise à jour date de 1994 sous le nom de MERISE/2 par Georges Panet et Raymond Letouche. Cette méthode et son MCD sont très largement diffusés en France.





## - Un peu d'histoire

# UML



Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un **langage** de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.

Les auteurs (Grady Booch, James Rumbaugh et Ivar Jacobson) ont mis en commun le résultat de leur recherche en matière de modélisation pour proposer non pas une méthode mais un langage commun.

Normalisé en 1997 puis régulièrement mis à jour (dernière version 2017) il propose 14 diagrammes pour décrire les SI, leur interaction et leur cycle de vie.

Parmi eux, le diagramme de classes qui s'inspire du modèle de Chen.

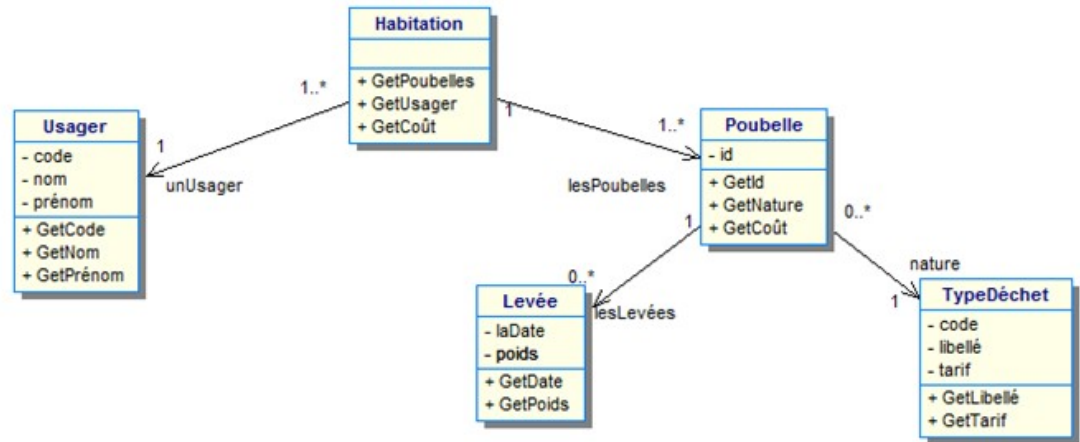
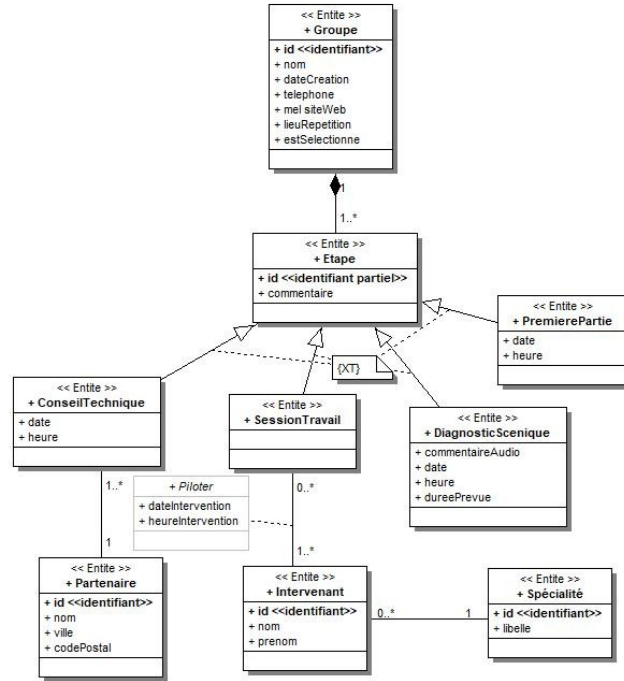


# Un peu d'histoire

# UML



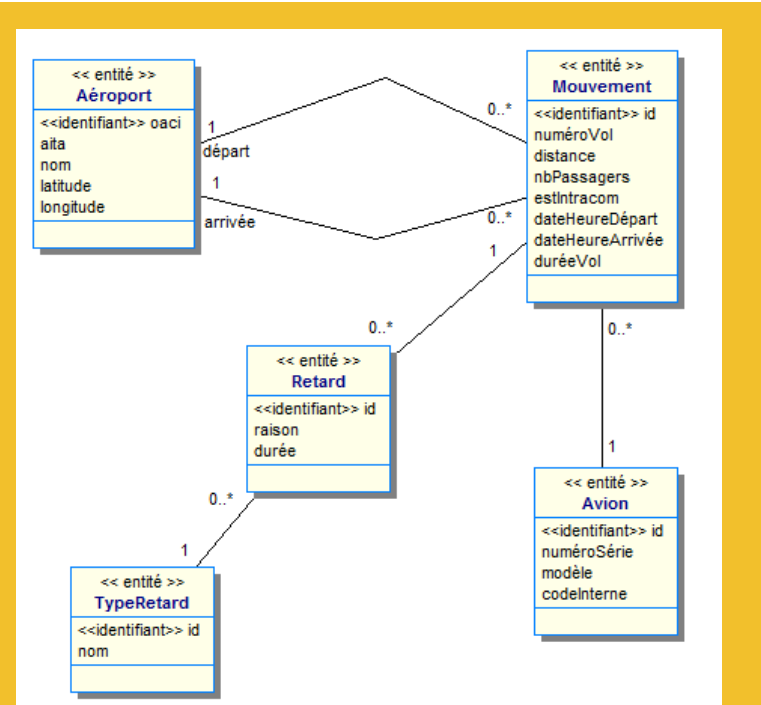
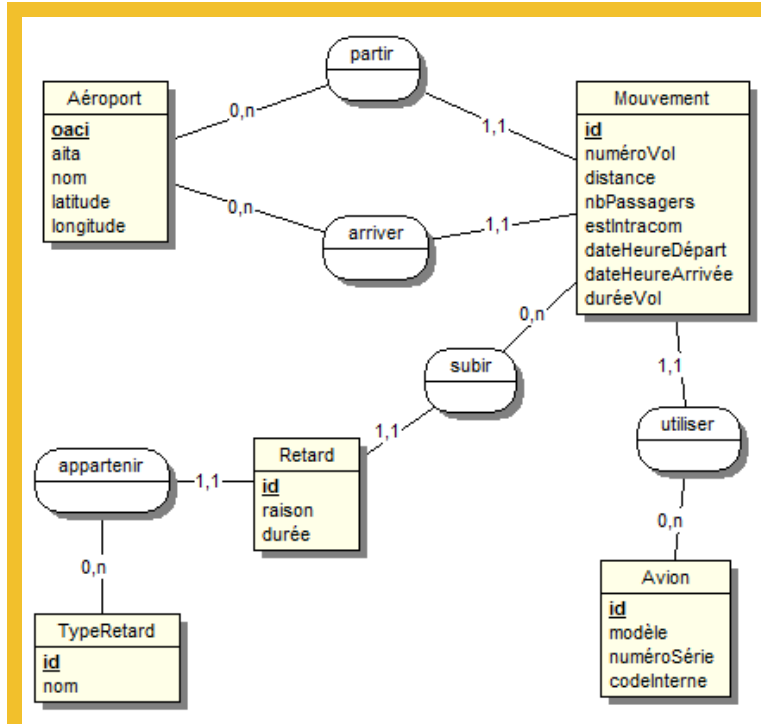
Le diagramme de classes peut être utilisé pour représenter une base de données mais aussi une conception d'objet en POO.



Source : différents sujets de BTS SIO



# MCD vs DdC





*Looping*



# Comment modéliser :



## Trois méthodes :

- par recherches des dépendances fonctionnelles
- par la recherche sujet/verbe/complément

*Une agence de voyage propose des croisières en bateau. Chaque croisière a un nom [...] et passe par [...] [différentes] escales correspondant à des villes [...].*

Pour gérer les réservations de ses clients, l'agence utilise une base de données.

Une croisière propose de une à dix escales. Chaque escale est spécifique à la croisière.

Une escale doit proposer des activités dont on ne proposera qu'une description sommaire ('Ballade de 3 km avec point de vue', 'Traverser d'un marché local', 'Visite de la basilique') et d'une durée approximative en heure. L'escale sera également décrite par sa durée en heure, la ville visitée, les villes à proximité avec le nombre de kilomètres pour s'y rendre.

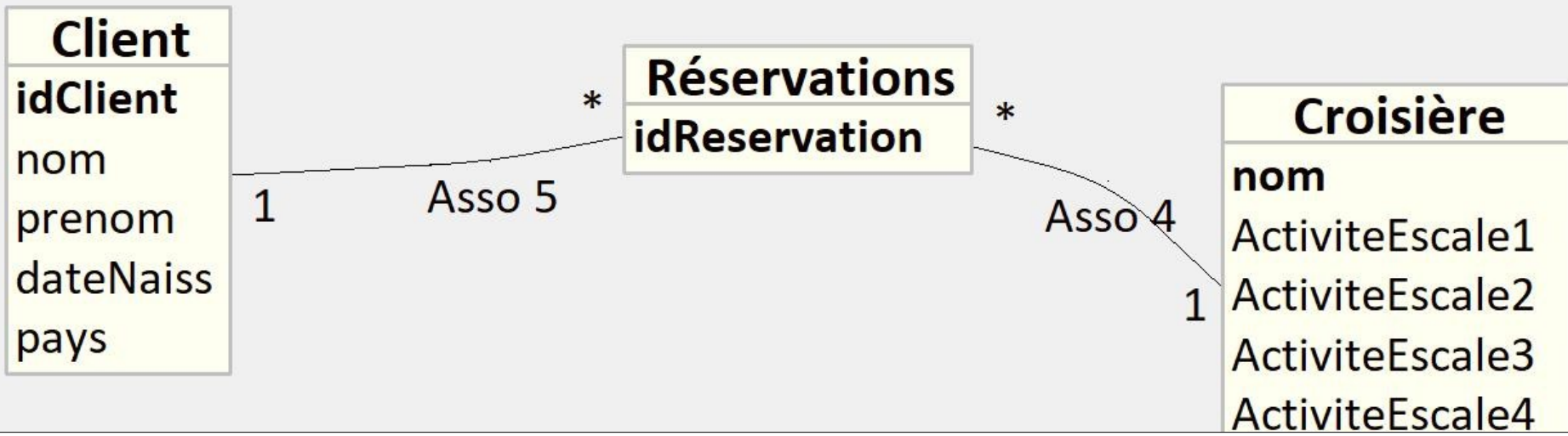
L'agence désire mémoriser pour chaque ville, le pays, la monnaie et un descriptif sous forme de texte des formalités nécessaire pour accoster.

Lorsqu'un client réserve pour une croisière, il doit préciser les nombres de personnes adultes et enfants qui l'accompagnent. Bien entendu, il ne peut acheter qu'une seule réservation par croisière. Un client sera caractérisé par son nom.





# - Cas Croisières

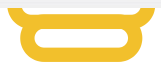


**Client**(idClient, nom, prenom, dateNaiss, pays)

**Réservations**(idReservation, #idClient, #nomCroisiere)

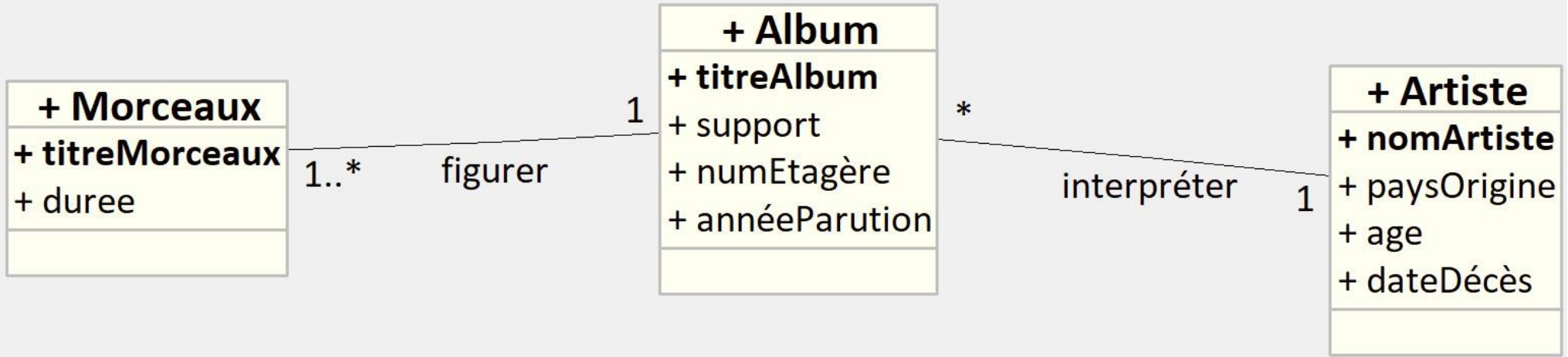
**Croisière**(nom, escale1, escale2, escale3, escale4, escale5, escale6, escale7, escale8, escale9, escale10, activitésEscale1, activitésEscale2... *pfff c'est long*)

**Ville**(nom, latitude, longitude, *euuh non...*)





## - Cas Gestion de CD



Album(titreAlbum, #nomArtiste, nbMorceau, durée, styleMusical, année, support, numEtagère)

Morceaux(titreMorceaux, durée, #titreAlbum)

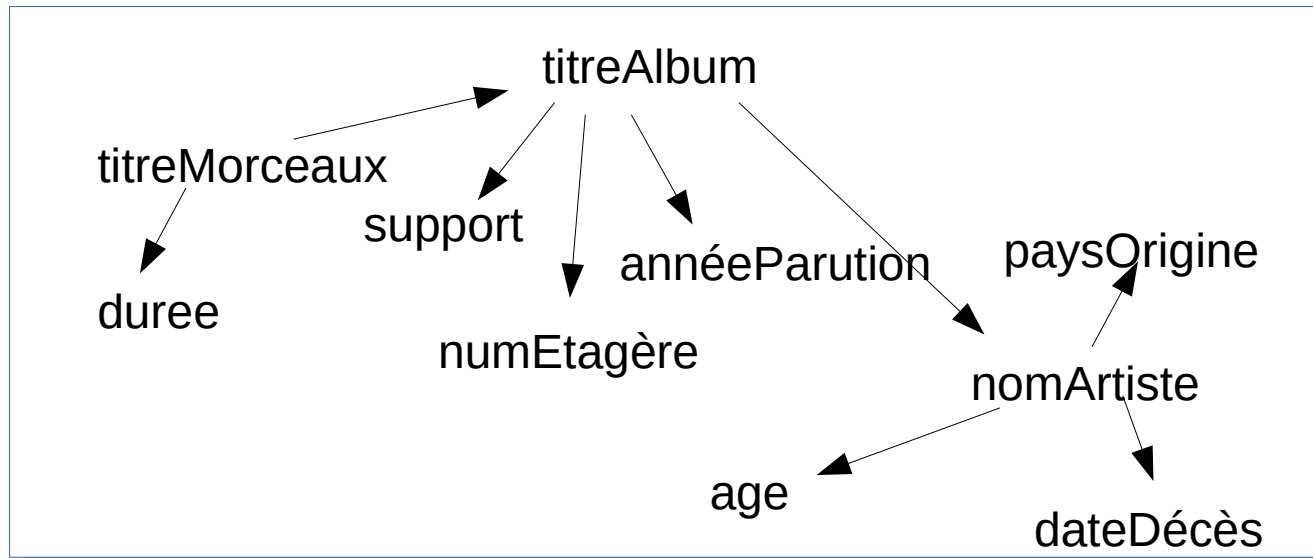
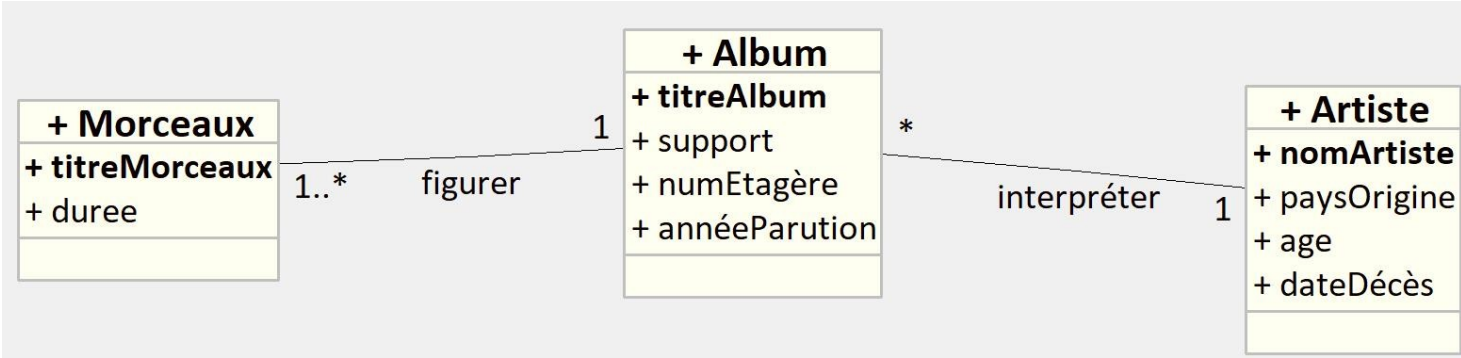
Artiste(nomArtiste, stylesMusical, paysOrigine, age, dateDécès)



# - Cas Gestion de CD

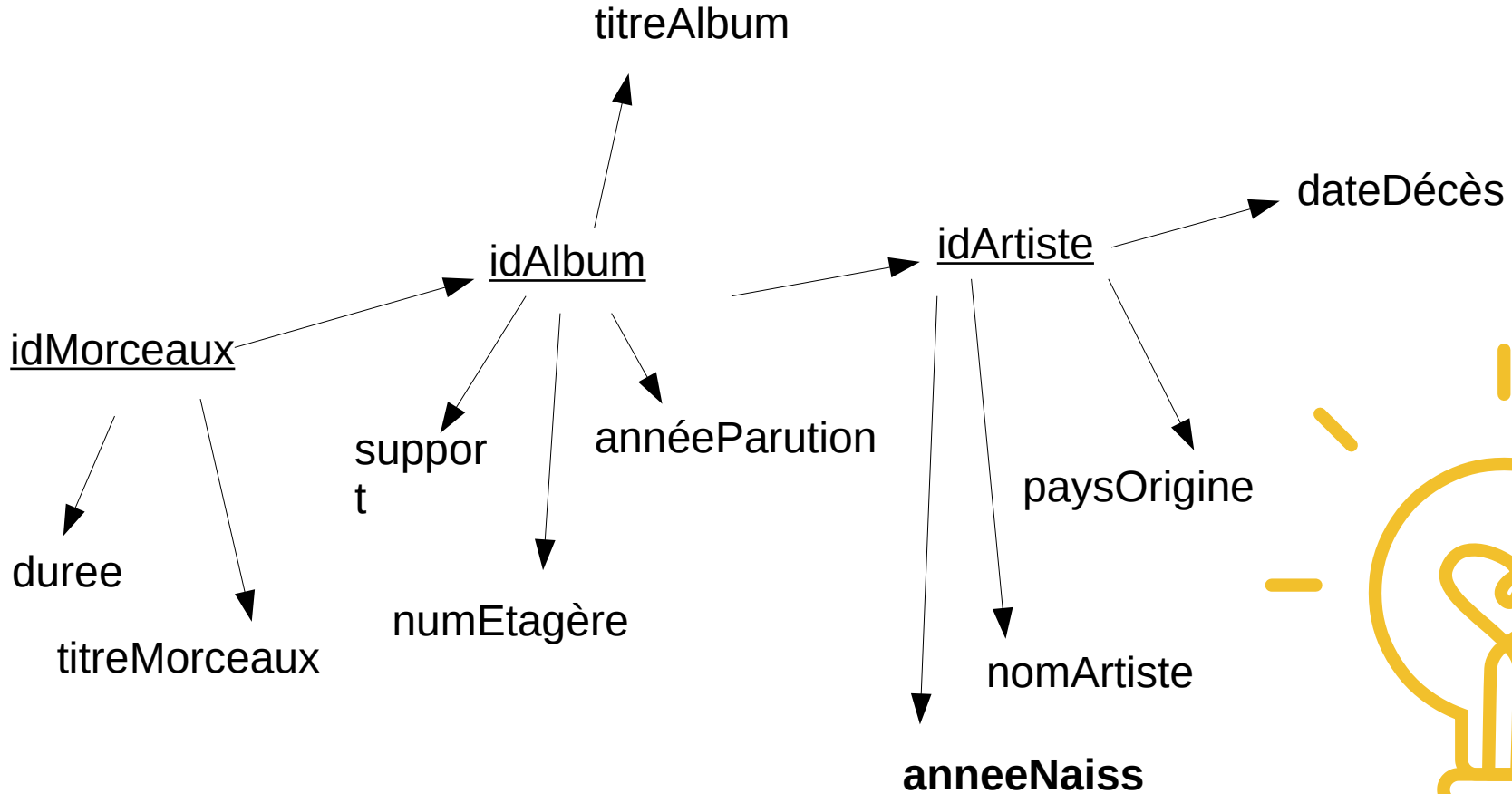


Une base de données est construite autour de la notion de dépendance fonctionnelle

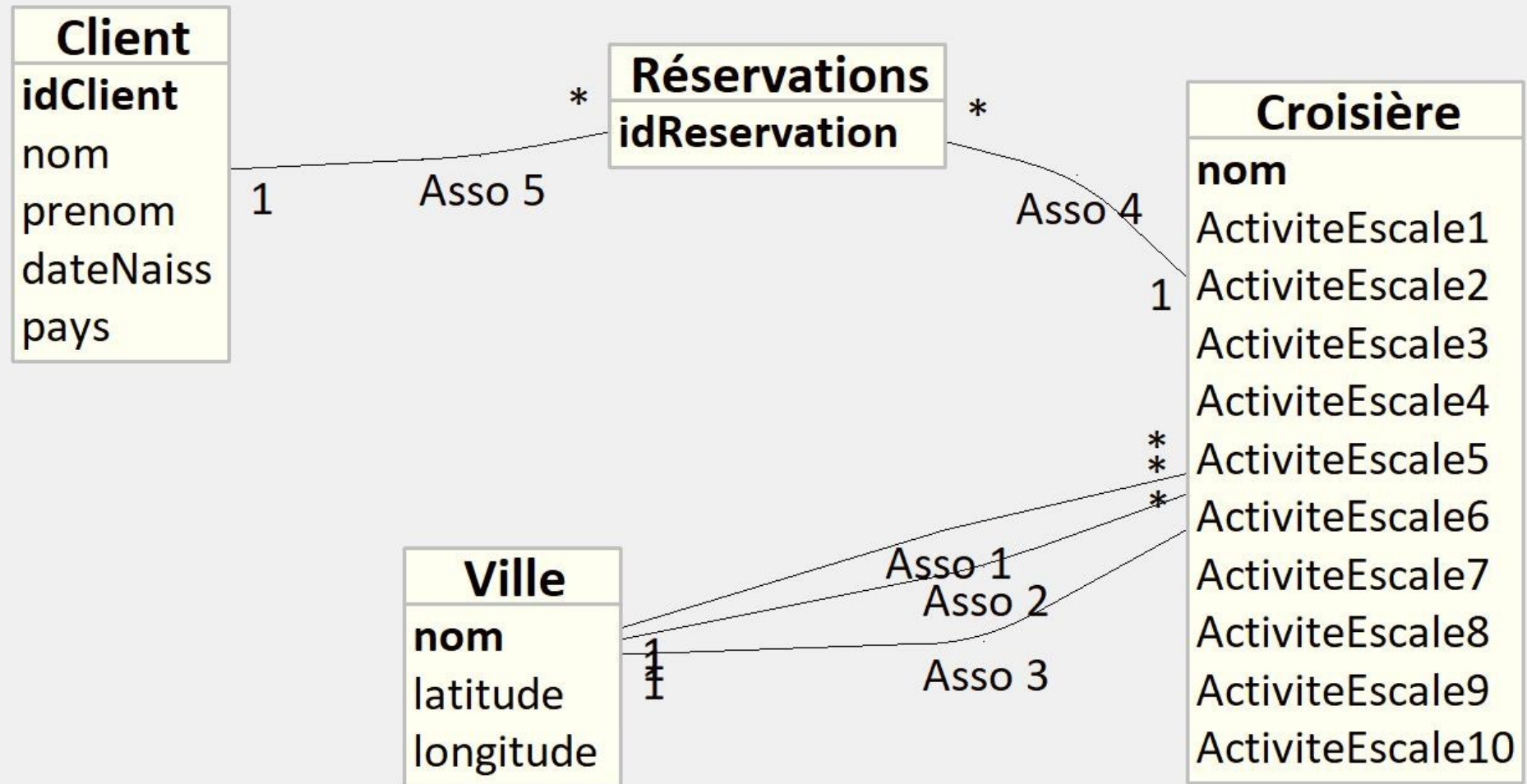


# - Cas Gestion de CD

En ajoutant des identifiants non ambigües :



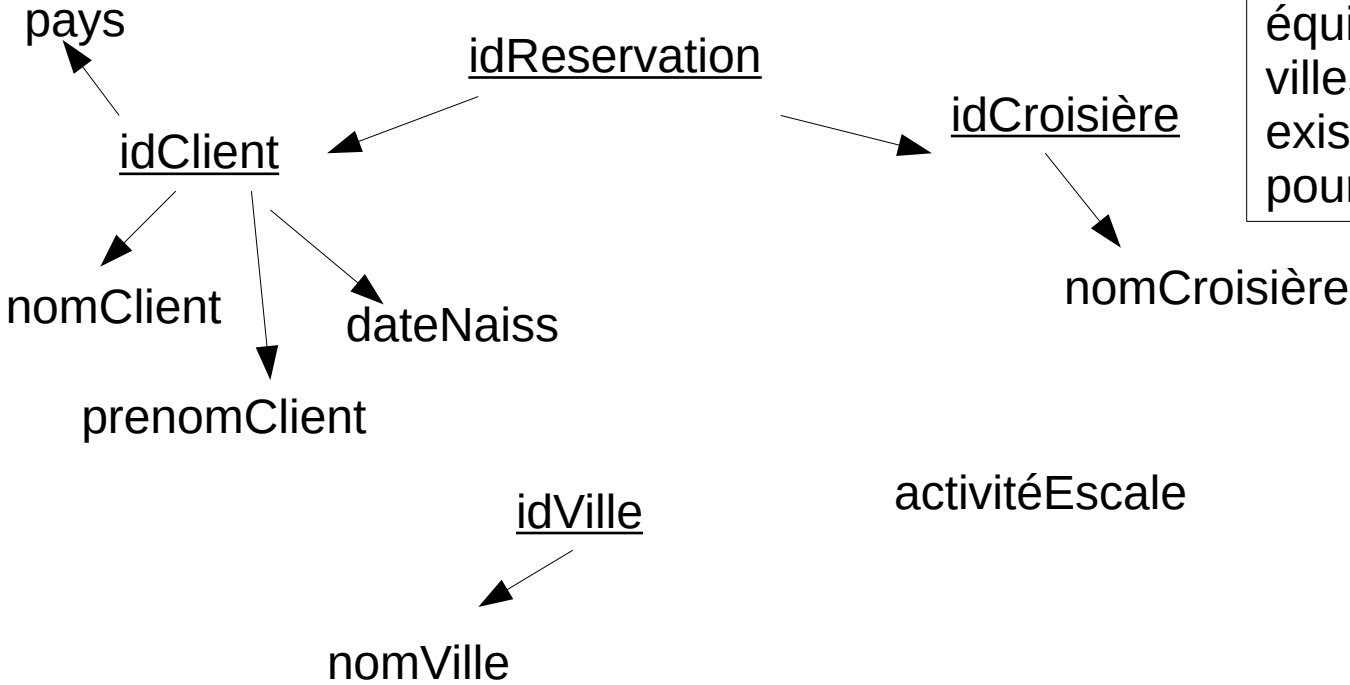
# - Cas Croisières



# - Cas Croisières

En ajoutant des identifiants non ambiguës :

ActivitéEscale1, activitéEscale2, activitéEscale3, ... sont en fait la même donnée dupliquée...



La relation entre Croisière et Ville pose un problème équivalent : il existe plusieurs villes pour une croisière et il existe plusieurs croisières pour la même ville.





# Au niveau des données

Au rythme des vagues	Activités
Marseille	Embarquement
Barcelone	Visite de 'La Sagrada Familia'
Valence	Visite arrière pays
Tunis	Concert de musique traditionnel
Alexandrie	Soirée dansante
Athènes	Visite historique
Naples	Visite historique

Au regard de ces données, les couples :

- ('Au rythme...', 'Marseille'),
- ('Au rythme...', 'Barcelone'),
- ('Au rythme...', 'Valence'),
- ('Au rythme...', 'Tunis'),
- ('Au rythme...', 'Alexandrie'),
- ('Au rythme...', 'Athènes'),
- ('Au rythme...', 'Naples').

Sont uniques.

Le couple (croisière, escale) est un bon identifiant.



# Au niveau des données

Au rythme des vagues	Activités
Marseille	Embarquement
Barcelone	Visite de 'La Sagrada Familia'
Valence	Visite arrière pays
Tunis	Concert de musique traditionnel
Alexandrie	Soirée dansante
Athènes	Visite historique
Naples	Visite historique
Marseille	Débarquement

Si une ville apparaît deux fois, il faut trouver un meilleur n-uplet pour fournir un identifiant.

Je vous laisse le trouver.



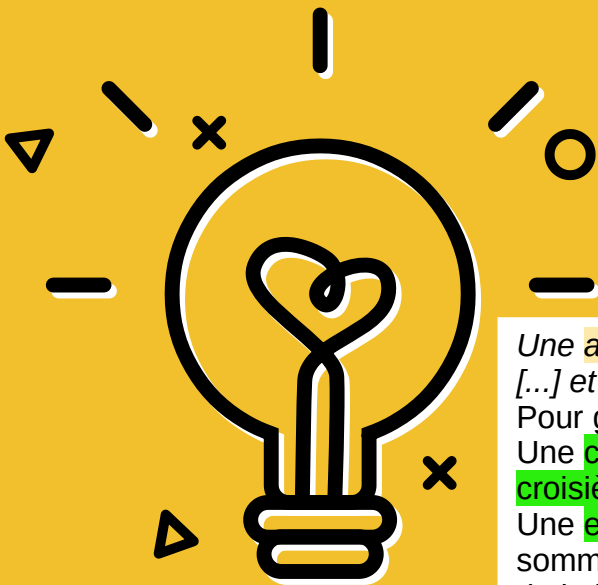


# Au niveau des données

Aventures et Rêveries	Activités
Marseille	Embarquement
Barcelone	Parcours Gaudi, Soirée dansante
Oran	Visite architecturale, Sur les traces de 'La Peste', Musée Ahmed zabana
Tanger	Soirée dansante
Arrecife	Carnaval

La croisière « Au rythme... » propose de une à trois activités par couple (croisière,escale)... en UML cela ne pose pas de problème.

# Comment modéliser :



## Trois méthodes :

- par recherches des dépendances fonctionnelles
- par la recherche sujet/verbe/complément

Une agence de voyage propose des croisières en bateau. Chaque croisière a un nom [...] et passe par [...] [différentes] escales correspondant à des villes [...].

Pour gérer les réservations de ses clients, l'agence utilise une base de données.

Une croisière propose de une à dix escales. Chaque escale est spécifique à la croisière.

Une escale doit proposer des activités dont on ne proposera qu'une description sommaire ('Ballade de 3 km avec point de vue', 'Traverser d'un marché local', 'Visite de la basilique') et d'une durée approximative en heure. L'escale sera également décrite par sa durée en heure, la ville visitée, les villes à proximité avec le nombre de kilomètres pour s'y rendre.

L'agence désire mémoriser pour chaque ville, le pays, la monnaie et un descriptif sous forme de texte des formalités nécessaire pour accoster.

Lorsqu'un client réserve pour une croisière, il doit préciser les nombres de personnes adultes et enfants qui l'accompagnent. Bien entendu, il ne peut acheter qu'une seule réservation par croisière. Un client sera caractérisé par son nom.

# Comment modéliser :

Classes candidates :  
Agence de Voyage  
Croisières  
Bateau  
Client  
Réservation  
Escale  
Ville  
Pays  
Formalités

Classes retenues :  
~~Agence de Voyage~~  
Croisières  
Bateau  
Client  
Réservation  
Escale  
Ville  
Pays  
Formalités

## es fonctionnelles complément

Une [...] Pour [...] Une cro [...] Une [...]

es en bateau. Chaque croisière a un nom correspondant à des villes [...].

agence utilise une base de données. Chaque escale est spécifique à la

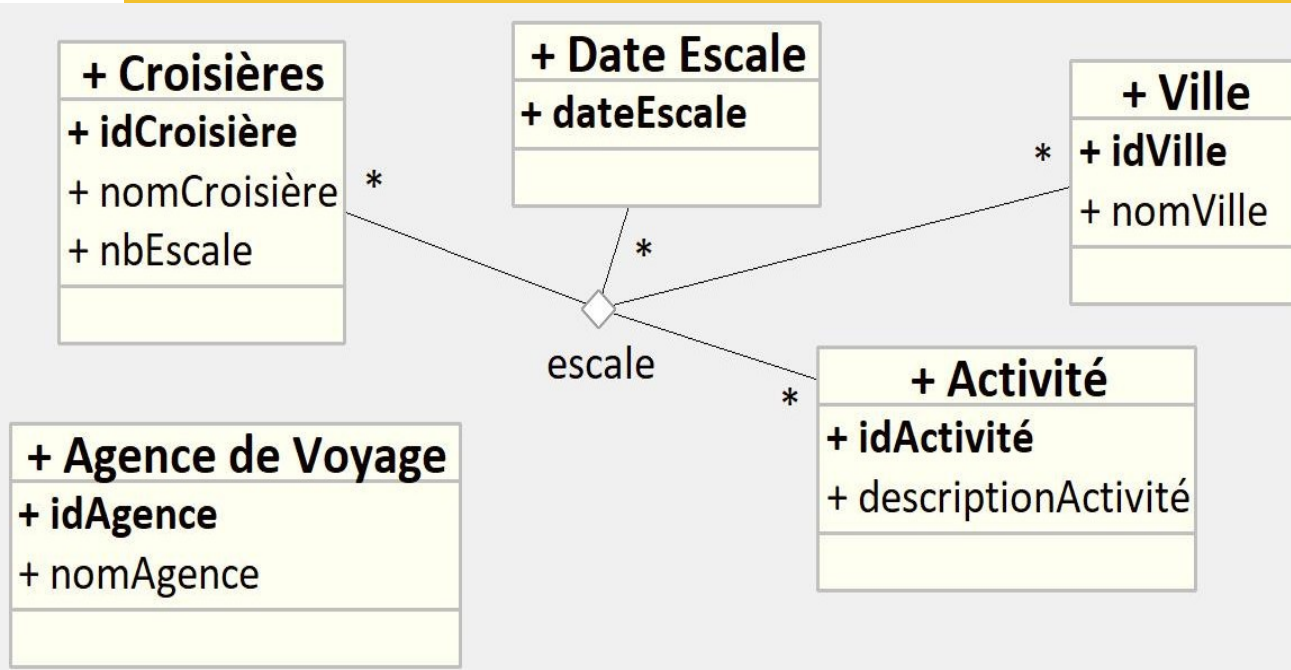
on ne proposera qu'une description sommaire ('Ballade de 3 km avec point de vue', 'Traverser d'un marché local', 'Visite de la basilique') et d'une durée approximative en heure. L'escale sera également décrite par sa durée en heure, la ville visitée, les villes à proximité avec le nombre de kilomètres pour s'y rendre.

L'agence désire mémoriser pour chaque ville, le pays, la monnaie et un descriptif sous forme de texte des formalités nécessaire pour accoster.

Lorsqu'un client réserve pour une croisière, il doit préciser les nombres de personnes adultes et enfants qui l'accompagnent. Bien entendu, il ne peut acheter qu'une seule réservation par croisière. Un client sera caractérisé par son nom.



# Erreurs fréquentes



- Association N-aire magique,
- Classe Date,
- Classe isolée,
- Classe du SI



# Erreurs fréquentes



- Association 1 – 1,
- Classe vide,
- Clé étrangère visible ou assoc cachée.

# Les Formes Normales des bases de données relationnelles

**01**

Une relation est en 1ère FN

- Si elle possède un identifiant.
- Si chaque attribut est monovalué.

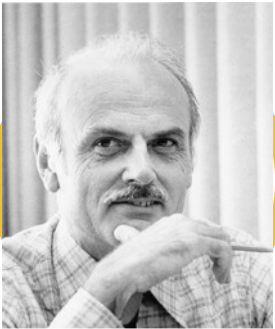
Une forme normale caractérise le fait que la structure de la base respecte certaines qualités.

- données non redondantes,
- intégrité des données.

**02**

Une relation est en 2e FN

- Si elle est en 1ère FN.
- Si chaque attribut non clé est en DF avec toute la clé.



Edgar Frank Codd 1923-2003

**03**

Une relation est en 3e FN

- Si elle est en 2e FN.
- Si chaque attribut non clé est en DF directement avec la clé.

Et

**04**

autres formes normales.

- la Boyce-Codd,
- la 4<sup>e</sup> FN,
- la 5<sup>e</sup> FN,
- la forme normale domaine clef.



Illustrations et CC :

- Yellow Idea Template for LibreOffice Impress by : @ealita.id, 2020.

- icones cerveau et machine :

 [https://thenounproject.com/creator/coquet\\_adrien/](https://thenounproject.com/creator/coquet_adrien/)

 Bibliographie :

- Georges Gardarin, Bases de Données

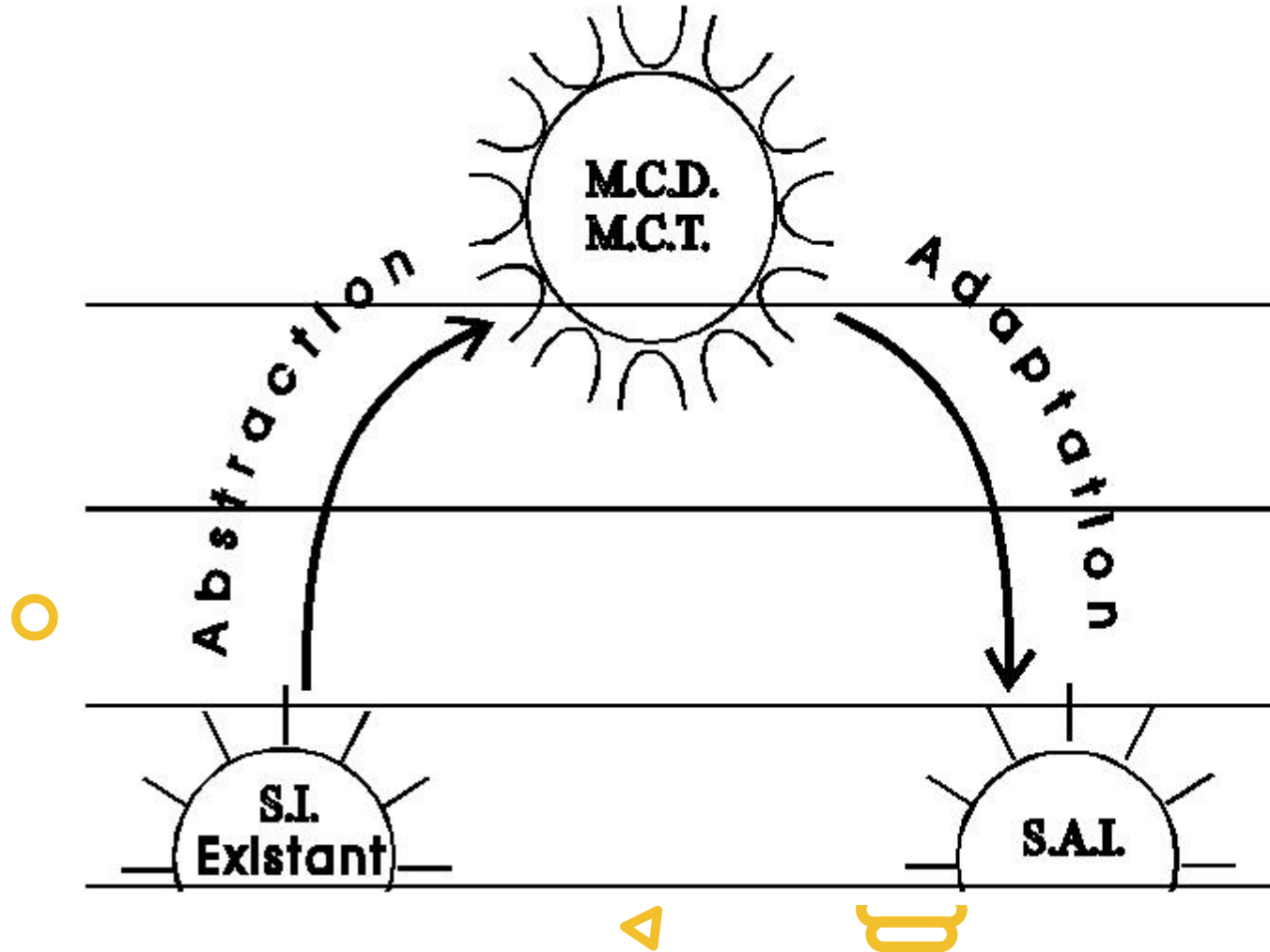
 - Christian Soutou, Modélisation des Bases de Données

- Hubert Tardieu, Arnold Rochfeld et René Colletti, La Méthode Merise. Principes et Outils, Editions d'Organisation

- Georges Panet et Raymond Letouche, Merise/2

 **Merci de votre attention**

# La courbe en soleil de Merise



Conceptuel

Organisationnel

Logique

Physique







# Propriétés d'un identifiant

(ou d'une clé primaire)

## Unique

C'est la première qualité attendue.

*Le numéro INSEE, N°carte d'électeur, N° d'une salle de classe, N° d'adhérent.*

## Constant

Immutabilité.

*Si le nom d'un artiste change, on ne peut pas utiliser ce champ pour identifiant.*

Si toutefois il arrive que l'identifiant change, il faut alors détruire l'enregistrement lié et en créer un nouveau.

## Existe (non NULL)

Difficile de définir quelque chose qui n'existe pas.

*Si le nom de l'album de Led Zeppelin n'a pas de nom, on ne peut pas utiliser ce champ pour identifiant des albums.*



Il appartient à un cadre de référence appelé le Système d'Information.

En dehors du SI, il peut ne pas avoir le même rôle. Ex n°tél.

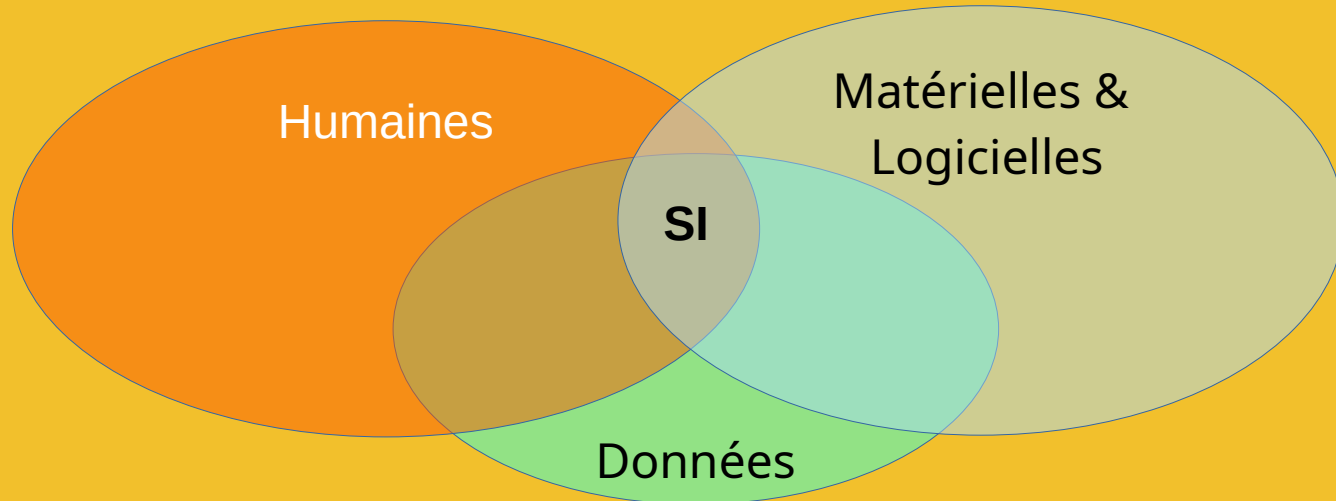




# Systeme d'information

est un ensemble organisé de ressources permettant d'acquérir, de traiter, de stocker et de diffuser de l'information dans et entre des organisations.

C'est l'intersection entre les ressources



# La dépendance fonctionnelle

C'est une propriété des données A et B, noté  $A \rightarrow B$ .

On dit que B est en dépendance fonctionnelle avec A, si la connaissance d'une valeur de A détermine précisément une et une seule valeur de B.

A sera la source de la DF, B sera la destination de la DF ou sera en dépendance fonctionnellement avec A.

idAlbum  $\rightarrow$  titreAlbum

idMorceaux  $\rightarrow$  dureeMorceaux

**DF issues par transitivité**

idMorceaux  $\rightarrow$  titreAlbum

idMorceaux  $\rightarrow$  nomArtiste

La dépendance fonctionnelle est transitive : Si  $A \rightarrow B$  et  $B \rightarrow C$  alors  $A \rightarrow C$ .

NB. : Les DFs issues de la transitivité sont à éviter. Les DFs non issues de transitivités sont dits **DF directe**.





# La dépendance fonctionnelle

C'est une propriété des données A et B, noté  $A \rightarrow B$ .

**On dit que B est en dépendance fonctionnelle avec A, si la connaissance d'une valeur de A détermine précisément une et une seule valeur de B.**

A sera la source de la DF, B sera la destination de la DF ou sera en dépendance fonctionnellement avec A.

La source de la DF peut être complexe :

~~idClient . idCroisière  $\rightarrow$  nomClient~~

idClient . idCroisière  $\rightarrow$  dateRéservation

~~idClient . IdCroisière  $\rightarrow$  nomCroisière~~

C'est une **DF élémentaire**



idClient  $\rightarrow$  nomCli

idCroisière  $\rightarrow$  nomCroisière

Lorsqu'une partie de la source est source d'une DF, ce n'est pas une DF intéressante.



# La dépendance fonctionnelle

C'est une propriété des données A et B, noté  $A \rightarrow B$ .

**On dit que B est en dépendance fonctionnelle avec A, si la connaissance d'une valeur de A détermine précisément une et une seule valeur de B.**

A sera la source de la DF, B sera la destination de la DF ou sera en dépendance fonctionnellement avec A.

Pour avoir une bonne base de données, toutes nos DF doivent être **élémentaires** et **directes**.

