

Base de Données

Polytech Lille - GBA4 - Semestre 7

1

0. Préambule

2

À propos... de moi



3

À propos... des supports

🕒 Inspirés de ceux d'Olivier Caron (MCF en IS), Anne Etien (PR à la FST), Walter Rudametkin (PR à Rennes)

🕒 Adaptation au volume horaire, à vos objectifs pédagogiques et (quand c'est possible) à votre discipline scientifique

4

À propos... du programme



5 séances de cours
5 séances de TP



1 contrôle TP **individuel** de 2h sur machine de TP



Système d'exploitation: linux
Tableur: Libre Office Calc
SGBD: Postgres

5

5

À propos... du contenu

1. Objectifs des bases de données **CM1**
2. Modélisation de systèmes d'informations **CM1** **TP1**
3. Tables dans une base de données relationnelles **CM2**
4. Requêtes **CM3** **CM4** **TP2** **TP3** **TP4** **TP5** **CTP**
5. Récapitulatif et Bilan **CM5**

6

6

À propos... de la bibliographie

- ▶ Introduction aux bases de données, *Serge Miranda et José-Maria Busta*
- ▶ Les bases de données relationnelles, *Serge Miranda et José-Maria Busta*
- ▶ Bases de données, *George Gardarin*
- ▶ Des Bases de données à l'Internet, *Philippe Mathieu*
- ▶ Ce support de cours 😊
- ▶ Exercices et examens précédents corrigés

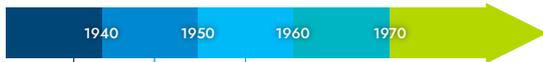
7

7

1. Objectifs des BD

8

Historique



Document papier
- structuration en fichiers, tableaux
- plusieurs utilisateurs mais vue unique des données

Structures de données en mémoire centrale
- stockage temporaire
- "petit" volume de données
- nécessite un langage de programmation
- contexte mono-utilisateur

Fichier informatisé
- stockage persistant sur disque
- "gros" volume de données
- langages de programmation ET systèmes de gestion de fichiers
- contexte mono-utilisateur (en général)

9

Historique — cas d'usage

Gestion des étudiants dans un école d'ingénieurs



Approche fichier:
- Information est conservée dans différents services
- Est-elle sous la forme souhaitée ?
- Les informations sont dupliquées. Sont-elles identiques ?

10

Historique — fichier informatique

- ▶ **Problèmes** de l'approche fichier:
 - ▶ Fichiers de nature **différente**, manipulés par des programmes **différents**, dans des langages **différents** avec des formalismes **différents**.
 - ▶ Difficulté à saisir les **liens** entre les données
 - ▶ Pas de **partage** de données entre les utilisateurs
 - ▶ **Centralisation** physique, mais pas logique.
 - ▶ **Dépendance** entre les données et les traitements
 - ▶ Absence de gestion de la **sécurité** des données
 - ▶ Pour toute nouvelle application, où sont les informations utiles ?

11

Historique



SGBD: Systèmes de Gestion de Base de Données
- Description **unique** et **globale** des données
- Langage **commun** de manipulation.
- **Centralisation** (pas de duplication, donc pas d'incohérence possible).
- **Indépendance** entre données et traitements
- Séparation des descriptions logique et physique des données
- **Contrôle** sémantique des données
- **Sécurité** (Qui a le droit de lire ou d'écrire, contrôle d'intégrité).
- **Partage** des données

12

Tableur vs Base de Données

	Tableur	BD
Utilisation Principale	Calculs	Gestion et Traitements des Données
Structuration des Données	Aucune	Structuration et Cohérence forte
Contrôle de l'intégrité des Données	Aucune*	Vérification Stricte des valeurs possibles pour chacune des données
Accès aux Données	Mono-utilisateur	Multi-Utilisateurs
Confidentialité des Données	Aucun contrôle	Vérification des droits d'accès pour chaque utilisateur
Taille des Données	1 table / quelques centaines de lignes	Plusieurs tables / Plusieurs milliers de lignes par table
Traitement sur les Données	Quantitatifs	Quantitatifs et Qualitatifs
Interrogation sur les Données	Procédures spécifiques	Langage "universel": SQL

13

13

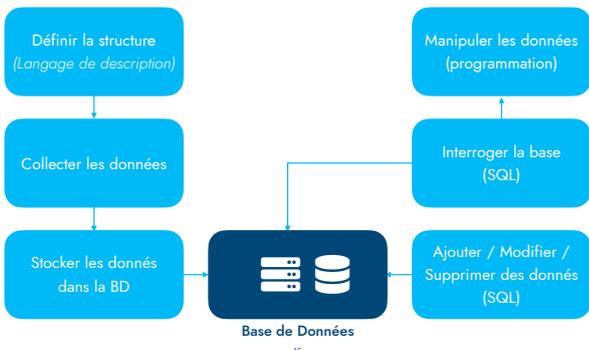
Tableur vs Base de Données

	Tableur en ligne	BD
Utilisation Principale	Calculs	Gestion et Traitements des Données
Structuration des Données	Aucune	Structuration et Cohérence forte
Contrôle de l'intégrité des Données	Aucune*	Vérification Stricte des valeurs possibles pour chacune des données
Accès aux Données	Multi-Utilisateurs	Multi-Utilisateurs
Confidentialité des Données	Vérification des droits d'accès pour chaque utilisateur	Vérification des droits d'accès pour chaque utilisateur
Taille des Données	1 table / quelques centaines de lignes	Plusieurs tables / Plusieurs milliers de lignes par table
Traitement sur les Données	Quantitatifs	Quantitatifs et Qualitatifs
Interrogation sur les Données	Procédures spécifiques	Langage "universel": SQL

14

14

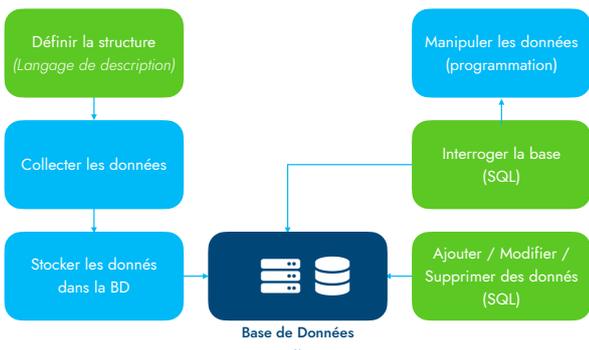
Cycle de vie d'une BD



15

15

Cycle de vie d'une BD



16

16

Propriétés des Bases de Données

1. **Indépendance** physique des données
2. **Indépendance** logique des données
3. Manipulation des données par des **non-informaticiens**
4. **Efficacité** des accès aux données
5. Administration **centralisée** des données
6. **Non-redondance** des données
7. **Cohérence** des données
8. **Partageabilité** des données
9. **Sécurité** des données

17

17

Indépendance Physique

- ▶ Représentation du monde réel sous forme indépendante de la structure physique
- ▶ En informatique, les utilisateurs de SGBD n'ont pas à se soucier de: la structure de stockage, fichiers, listes, tableaux, disque dur, système de sauvegarde
- ▶ Le point d'entrée est la connexion au serveur de base de données sans connaître les détails de l'infrastructure

18

18

Indépendance Logique

- ▶ **Idée** : chaque utilisateur doit pouvoir ne connaître qu'une partie de la sémantique des données et ne voir qu'une partie des données
- ▶ **Avantages** :
 - ✓ On peut modifier (virtuellement) la structure
 - ✓ On ne dispose que des informations pertinentes
 - ✓ Sécurité (on ne peut pas accéder à des données qu'on ne connaît pas).
 - ✓ Plusieurs représentations sont possibles destinées à un utilisateur ou un groupe d'utilisateurs

19

19

Manipulation des données

- ▶ Les utilisateurs peuvent :
 - ▶ consulter les données
 - ▶ les mettre à jour
- ▶ Il faut donc définir un langage le plus simple et le plus intuitif possible :
 - ▶ **QBE** (Query By Example) : simple pour des requêtes simples (graphique)
 - ▶ **SQL** (Structured Query Language) : standard de fait, très puissant

20

20

Manipulation des données

- ▶ **QBE** (Query By Example) : simple pour des requêtes simples (graphique)

Query by example (QBE) grid

Dynaset created when query is run

Client	First name	Last name	Address
Client ID	John	Smith	123 main
First name	Mark	Johnson	420 street
Last name	Jen	Griffen	500 street
Address			
City			
State			
Zipcode			

Field:	Firstname	Lastname	Address	City
Table:	Client	Client	Client	Client
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				

21

21

Manipulation des données

- ▶ **SQL** (Structured Query Language) : standard de fait, très puissant

```
SELECT nom, num_departement, population
FROM communes
ORDER BY population DESC
LIMIT 10;
```

22

22

Efficacité de l'accès

- ▶ Réservé aux utilisateurs informaticiens :
- ▶ Offrir un langage de programmation et une bibliothèque pour la gestion des BD
 - ▶ Propriétaires: **PL-SQL** (Oracle), **VBA** (Microsoft), **libpostgres** (C), php...
 - ▶ Standards: **ODBC** (Microsoft), **JDBC** (Sun JavaSoft)
- ▶ Algorithmes d'optimisation des requêtes pour minimiser la charge du réseau

23

23

Efficacité de l'accès

- ▶ Exemple de fichier **PHP**

```
<?php
$dbconn = pg_connect(
    "host=localhost
    dbname=publishing
    user=www
    password=foo")
    or die('Connexion impossible : '
        . pg_last_error());

$query = 'SELECT * FROM authors';
$result = pg_query($query)
    or die('Échec de la requête : '
        . pg_last_error());
```

24

24

Administration cohérente

- ▶ L'administration est souvent **centralisée**
- ▶ Le rôle de l'administrateur:
 - ▶ Contrôle efficace des données, sauvegarde
 - ▶ Résoudre les conflits
 - ▶ Optimiser les accès
 - ▶ Prise en compte de la charge (nombre d'utilisateurs, charge réseaux et des serveurs / disques)

25

25

Non redondance des données

- ▶ Éviter la perte de place en mémoire
- ▶ Éviter les mises à jour multiples

```
Personne(nom, prenom, age, profession)
Habitant(nom, prenom, numero, rue, code_postal, ville)
```

- ▶ Problèmes ? Duplication (nom, prenom), cohérence pour les mises à jour

26

26

Non redondance des données

- ▶ Éviter la perte de place en mémoire
- ▶ Éviter les mises à jour multiples

```
Personne(ID, nom, prenom, age, profession)
Habitant(ID, numero, rue, code_postal, ville)
```

27

27

Cohérence des données

- ▶ Notion d'intégrité
 - ▶ de domaine, référentielle,...

```
CREATE TABLE produits (
  no_produit INTEGER,
  nom TEXT,
  prix NUMERIC CHECK (prix > 0),
  prix_promotion NUMERIC CHECK (prix_promotion > 0),
  CHECK (prix > prix_promotion)
);
```

28

28

Cohérence des données

- ▶ Notion d'intégrité
 - ▶ de domaine, référentielle,...
- ▶ Fournir des outils de contrôle automatique

```
SellEndDate >= SellStartDate OR SellEndDate IS NULL;  
zip LIKE '[0-9][0-9][0-9][0-9][0-9]';
```

29

29

Partageabilité des données

- ▶ Permettre aux applications de partager les données de la base dans le temps mais aussi **simultanément**.
- ▶ L'utilisateur n'a pas à se soucier des autres accès
- ▶ Systèmes de **transactions** pour garantir que la base de données est dans un état cohérent (éviter ajout / modifications / suppressions sur une même donnée simultanément)

30

30

Sécurité des données

- ▶ Fournir des mécanismes pour définir des droits d'accès.
- ▶ Les données doivent être protégées contre les accès non autorisés ou mal intentionnés.
- ▶ Les droits d'accès peuvent également dépendre de la valeur des données
 - ▶ *Exemple* : un employé pourra connaître les salaires des personnes qu'il dirige mais pas des autres employés de l'entreprise.

31

31

Autres qualités d'un SGBD

- ▶ Gérer de grandes quantités de données (data-warehouse)
- ▶ Gérer des données réparties sur plusieurs serveurs
- ▶ Accepter un grand nombre de connexions
- ▶ Exécuter des requêtes avec un temps de réponse le plus court possible
- ▶ Proposer des mécanismes de reprise après panne (dans un état cohérent), système de réplication,...
- ▶ Avoir une bonne ergonomie
- ▶ Être à un prix abordable

32

32

Les SGBDR (relationnelle)

- Représentent 80% du marché
- Dimensionnés pour les besoins de la taille d'une entreprise
- Les Pros : Oracle, Informix, Sybase, DB2
- Les semi-pros : Postgres, MS-SQL server...
- et les autres : Ms Access, MySQL, MariaDB

33

33

Les SGBD non-relationnelle

- Pour des besoins très spécifiques: très gros volume de données (Amazon, Google), infrastructure très distribuée, focalisation sur la performance au détriment de la cohérence
- Exemple: *Cassandra*, *MongoDB*, *CouchDB*

```
{
  "_id": ObjectId("4efa8d2b7d284dad101e4bc7"),
  "Nom": "PELLERIN",
  "Prénom": "Franck",
  "Âge": 29,
  "Adresse": {
    "Rue": "1 chemin des Loges",
    "Ville": "VERSAILLES"
  }
}
```

34

34

2. Modélisation

35

Méthodologie et Modélisation

1. Modélisation : **description structurelle**
2. Méthodologie : démarche de conception
3. Certaines méthodes sont liées à une notation: *OMT*, *Merise* (MCD)
4. La notation *UML* indépendante de toute méthodologie

36

36

Le modèle Entité - Association

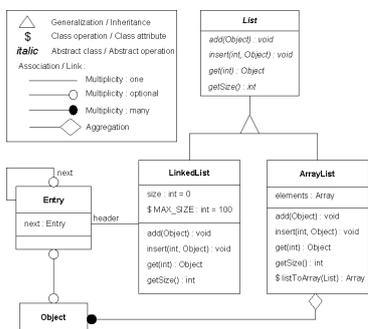
37

- ▶ Abréviation: **modèle E-A**
- ▶ Issu des travaux de Chen (US) et Européens
- ▶ Objectifs :
 - ▶ **Puissance** de représentation
 - ▶ **Stabilité** et **flexibilité** : un ajout de données ne doit pas remettre en cause le schéma
 - ▶ **Simplicité** : facilité de compréhension et d'utilisation
 - ▶ **Indépendance** par rapport à l'implémentation cible (SGBDR, fichiers, programmation...)
- ▶ Plusieurs réalisations : **MCD** (Merise) , **OMT** (+méthodologie), **UML**,...

37

OMT (Object Modeling Technique)

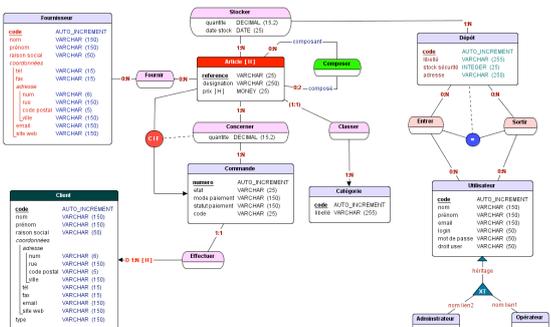
38



38

MERISE MCD*

39

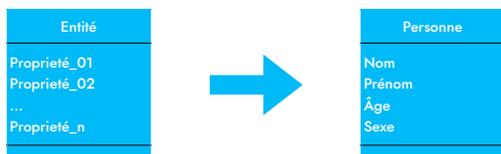


39

Principes de modélisation

40

- ▶ **Classe**:
Une classe est un **concept identifiable** et pertinent pour l'application, par exemple, une personne, une voiture, ... mais qui peut être aussi un concept ou une grandeur abstraite.
- ▶ La classe est définie par une **liste de propriétés** qui la caractérisent. Celles-ci constituent le plus petit élément d'information ayant un sens par lui-même.



40

Principes de modélisation

▶ Classe vs Objet:

- ▶ On distingue les notions de classe et objet
- ▶ On ne s'intéresse pas à chaque objet séparément, mais à un type d'objet

Personne	P1:Personne
Nom	Dequidt
Prénom	Jeremie
Âge	42
Sexe	M

▶ Classe:

Une classe est l'ensemble de tous les objets qui sont de la même nature et jouent le même rôle.

41

41

Principes de modélisation

▶ Identifiant:

Une des propriétés d'une classe, doit permettre d'identifier une de ses occurrences parmi toutes les autres.

- ▶ On parle également de **clé** (ex: numéro de sécurité sociale, login machine de TP...)

Personne	Personne	Personne
num_secu [clé]	num_secu <<identifiant>>	num_secu
Nom	Nom	Nom
Prénom	Prénom	Prénom
Âge	Âge	Âge
Sexe	Sexe	Sexe

Identifiant = num_secu

42

42

Principes de modélisation

▶ Terminologie UML:

- ▶ Entité type → Classe
- ▶ Entité → Instance / Objet
- ▶ Propriété → Attribut
- ▶ Identifiant → n'existe pas par défaut en UML

43

43

Principes de modélisation

▶ Association:

Une association est un lien logique entre l'ensemble des occurrences de classes éventuellement différentes.

- ▶ Une association est souvent perçue comme une action menée vis à vis de classes (se traduit par un verbe).
- ▶ La mise en relation de classes peut faire **apparaître des propriétés** qui n'appartiennent en propre à aucune des classes. On distingue :
 - ▶ Les associations **binaires** relient les différentes instances de deux classes d'entité (imposé par OMT et ODMG)
 - ▶ Les associations **n-aires** relient les instances de n-classes d'entité.
 - ▶ Les associations **réflexives** relient une instance d'une classe à d'autres instances d'une même classe.

44

44

Principes de modélisation

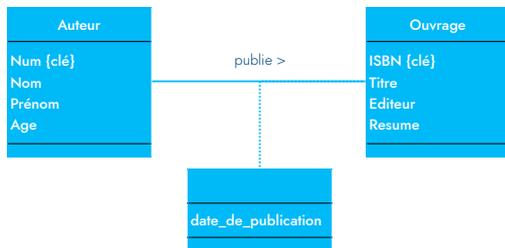
Association binaire:



45

Principes de modélisation

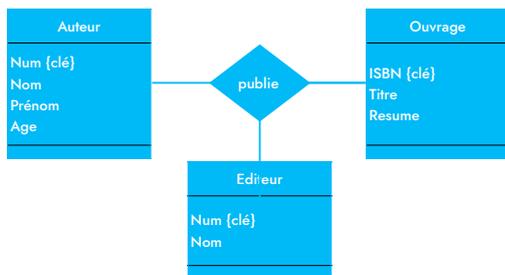
Association binaire (avec propriétés d'association):



46

Principes de modélisation

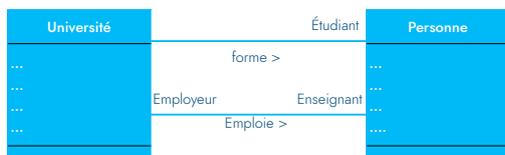
Association n-aire:



47

Principes de modélisation

Rôles d'une association:

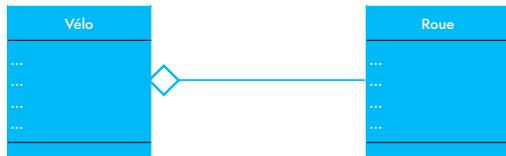


48

Principes de modélisation

► Agrégation:

Relation non symétrique qui exprime un couplage fort et une relation de type maître-esclaves
Une des classes joue un rôle plus important que l'autre dans la relation



Si le vélo est détruit, les roues peuvent être réutilisées

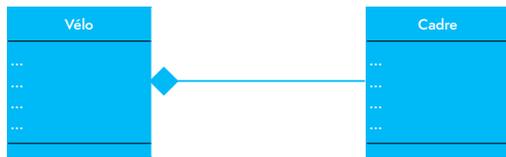
49

49

Principes de modélisation

► Composition:

Relation non symétrique qui exprime un couplage fort et une relation de type maître-esclaves
Une des classes joue un rôle plus important que l'autre dans la relation
Il ne peut y avoir qu'un **seul** maître.
Classes composites physiquement contenus par l'agrégat (inclusion des cycles de vie). **Si l'agrégat est détruit ses composants le sont aussi.**



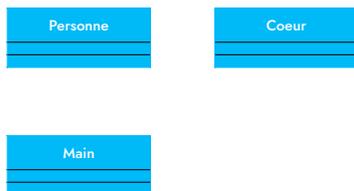
Si le vélo est détruit, le cadre aussi

50

50

Principes de modélisation

► Différence composition / agrégation:

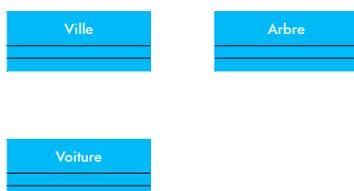


51

51

Principes de modélisation

► Différence composition / agrégation:



52

52

Principes de modélisation

Navigation:

Forme particulière d'association

Permet de se déplacer dans le modèle

Correspond aux différents scénarios (dynamique)

Par défaut, les associations sont navigables dans les deux directions



53

53

Principes de modélisation

Cardinalités:

Elles précisent les nombres minimum et maximum d'occurrences d'une entité pouvant être impliquées dans une occurrence de l'association.

Elles sont définies au niveau de **chaque extrémité** de l'association.

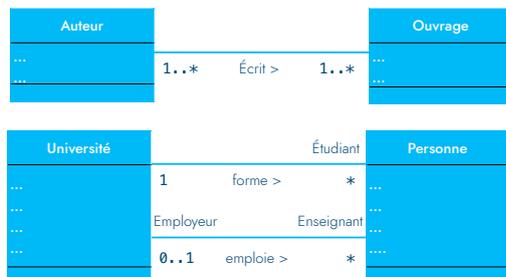
représentation	signification
1	Un et un seul
0..1	Zéro ou un
M..N	De M à N
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	De un à plusieurs

54

54

Principes de modélisation

Cardinalités:



55

55

Autres possibilités d'UML

- ▶ Des types pour les propriétés (attributs)
- ▶ Des paquetages
- ▶ Des droits d'accès (public, privé, etc)
- ▶ De l'héritage
- ▶ D'autres diagrammes

56

56

UML et BD Relationnelles

- ▶ Tout n'est pas obligatoire
- ▶ Trouver le bon compromis !
- ▶ Surcharge du schéma, difficulté de lecture
 - ▶ Parfois **KISS** (*Keep It Stupid Simple*) est une bonne approche
- ▶ Tout n'est pas possible facilement (exemple : héritage)
- ▶ La solution : définir un profil UML

57

57

Mise en pratique

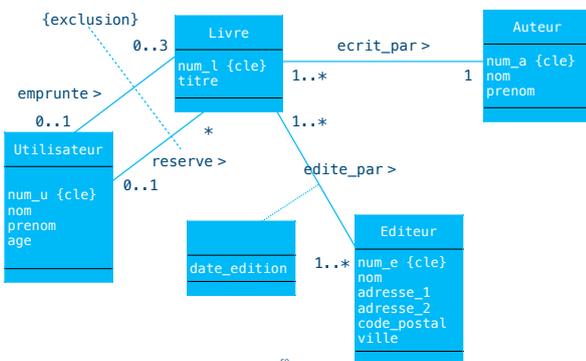
- ▶ Soit le cahier des charges suivants **volontairement flou** pour gérer une bibliothèque.

“ Un livre est caractérisé par son titre, son auteur, ses éditeurs. Pour chaque livre édité, on veut connaître la date d'édition. Un auteur est caractérisé par son nom, prénom. Un éditeur est caractérisé par son nom et son adresse. Un utilisateur est caractérisé par son nom, prénom et âge. Le but de cette gestion est de gérer les emprunts de livres (maximum 3 livres par utilisateur) et la possibilité de réserver des livres.

58

58

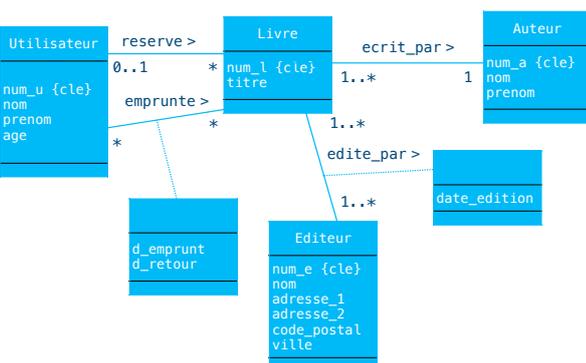
Une solution possible



59

59

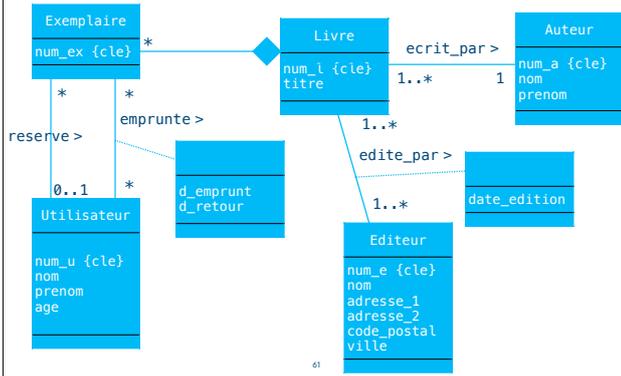
Une autre solution (historique des emprunts)



60

60

Une autre solution (plus réaliste)



61

Agence de voyage

- ▶ Informatisation de la gestion d'une agence de voyages comprenant :
 - ▶ les demandes d'informations sur les voyages,
 - ▶ les réservations de places pour un voyage,
 - ▶ la publicité (envoi du catalogue des voyages) auprès des clients des 3 dernières années.
- ▶ Un voyage est décrit par le (ou les) nom(s) du (des) pays visités et pour chaque jour :
 - ▶ la ville étape,
 - ▶ le nombre de km parcouru,
 - ▶ l'hôtel d'accueil.
- ▶ Pour certains pays des vaccinations sont obligatoires. Pour en informer ses clients, l'agence mémorise par pays les noms des vaccins et les délais de vaccination (entre la prise du vaccin et le début du voyage).

62

Agence de voyage

63
