






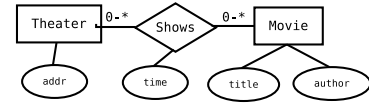
Traduction du modèle E/A en relations

Composé avec \LaTeX , révision 3997.

1

Modélisation entité-association (E/A)

-  Entité objet identifié comme tel (**nom**)
-  Association lien entre 2 entités (**verbe**), éventuellement réflexif
-  Attribut propriété d'une entité ou association
- cardinalité** dénombrement des liens entre entité ? 1 * +

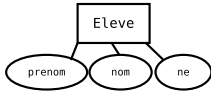


2

Traductions des concepts

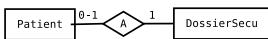
attribut en attribut !

entité en relation, avec ses attributs



Eleve(prénom **TEXT**,
nom **TEXT**,
ne **DATE**);

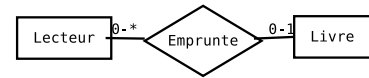
association card max 1-1 fusion (sauf multiple ?)



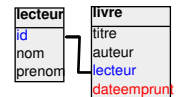
PatientDossierSecu(
nom **TEXT**, prénom **TEXT**, ne **DATE**,
secuid **TEXT**, nactes **INTEGER**);

3

association card max 1-n clef primaire côté n, clef étrangère côté 1
attributs de l'association ajouté à l'entité côté 1



Lecteur
(id **INTEGER PRIMARY KEY**,
nom **TEXT**, prénom **TEXT**);
Livre
(titre **TEXT**, auteur **TEXT**,
lecteur **INTEGER REFERENCES** Lecteur,
dateemprunt **DATE**);

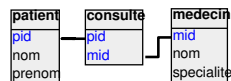


4

association card max n-n clefs primaires et table de liaison
dans ce cas, l'association est une relation



Patient(
pid **INTEGER PRIMARY KEY**,
nom **TEXT**, prénom **TEXT**);
Medecin(
mid **INTEGER PRIMARY KEY**,
nom **TEXT**, specialite **TEXT**);
Consulte(
pid **INTEGER REFERENCES** Patient,
mid **INTEGER REFERENCES** Medecin);

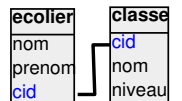


5

cardinalités minimales contraintes sur les attributs
0 attribut peut être **NULL**, 1 attribut **NOT NULL**



Classe(
cid **INTEGER PRIMARY KEY**,
nom **TEXT**, niveau **TEXT**);
Ecolier(
nom **TEXT**, prénom **TEXT**,
cid **INTEGER NOT NULL REFERENCES** Classe);



6

Conseils

- faire le dessin !** éviter tables directes
sinon hypothèses implicites de cardinalité...
- attributs** avec un nom explicite, un type bien choisi
n noSecu, nom nomClient, ad adresseClient
- utiliser les contraintes** sur les colonnes
NULL, NOT NULL, UNIQUE, CHECK...
- clef primaire simple** déclarée sur toutes les entités
nom de l'attribut unique dans la base (jointures naturelles)
numérotation automatique avec **SERIAL**
Specialite(
sid **SERIAL PRIMARY KEY**,
description **TEXT UNIQUE NOT NULL**);

7

déclarer les clefs étrangères entières simples
même nom d'attribut et même type que la clef primaire visée
simples, sinon duplications des informations !

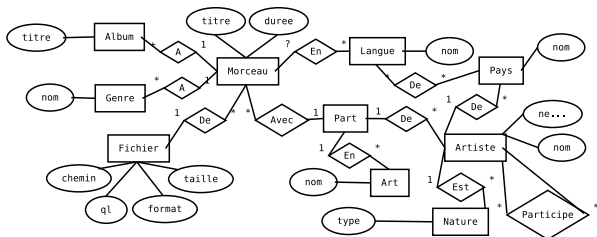
```
Medecin(  
mid SERIAL PRIMARY KEY,  
nom TEXT NOT NULL,  
sid INTEGER REFERENCES Specialite);
```

tables de liaisons sans clef primaire simple si inutile ?
unicité de la liaison, clef primaire composite

```
Consulte(  
pid INTEGER NOT NULL REFERENCES Patient,  
mid INTEGER NOT NULL REFERENCES Medecin,  
PRIMARY KEY (pid,mid));
```

8

Traduire le schéma Musique



9

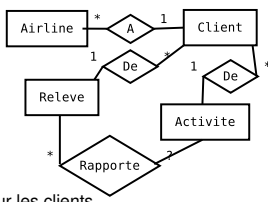
Erreur type clef étrangère

```
Fruit(fid INT8 PRIMARY KEY);
Arbre(aid INT8 PRIMARY KEY,
      fid INT4 REFERENCES Fruit);

-- INT8 et INT4 incompatibles vers 2^31 soit 2.10^9 environ
```

10

Propagation d'identité : le triangle maudit



- airline** compagnies aériennes
- client** comptes points fréquence pour les clients
 - situation des points à une date
- activite** l'activité du client à pour conséquence :
 - ajouts de points, consommation de points pour des cadeaux
- releve** mensuel notifié par courrier

11

Traduction initiale en relations

— pas de clef primaires simples, donc clef étrangères multiples

```
Airline
Client
  (airID INTEGER
   REFERENCES Airline,
   noCompte VARCHAR(16),
   solde INTEGER NOT NULL,
   dateSolde DATE NOT NULL,
   PRIMARY KEY(airID, noCompte));

Releve(
  airID INTEGER REFERENCES Airline,
  noCompte VARCHAR(16),
  dateReleve DATE,
  soldeReleve INTEGER NOT NULL,
  PRIMARY KEY(airID, noCompte, dateReleve),
  FOREIGN KEY(airID, noCompte) REFERENCES Client);
```

12

```
Activite
(activiteID INTEGER PRIMARY KEY,
 montant INTEGER NOT NULL,
 dateActivite DATE NOT NULL,
 -- le compte concerne
 airID1 INTEGER NOT NULL,
 noCompte1 VARCHAR(16) NOT NULL,
 FOREIGN KEY(airID1, noCompte1) REFERENCES Client,
 -- le releve dans lequel il apparait
 airID2 INTEGER NULL,
 noCompte2 VARCHAR(16) NULL,
 dateReleve DATE NULL,
 FOREIGN KEY(airID2, noCompte2, dateReleve)
 REFERENCES Releve);
```

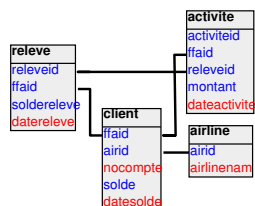
13

Conséquences

- duplications des informations
 - compagnie aérienne, client, date de relevé...
- deux clef étrangères presque identiques!
 - dans Activite vers Releve et Client
- fusion de deux compagnies? (Air France, KLM)
 - modifications de toutes les lignes de toutes les tables!
- complexité des requêtes, jointures multi attributs...

14

Clefs primaires simples : la bonne solution



```
Airline
(airID INTEGER PRIMARY KEY,
 airlineName VARCHAR(32)
 NOT NULL UNIQUE);
```

15

```
Releve
(releveID INTEGER PRIMARY KEY,
 ffaID INTEGER NOT NULL REFERENCES Client,
 soldeReleve INTEGER NOT NULL,
 dateReleve DATE NOT NULL,
 UNIQUE (ffaID, dateReleve));
```

```
Activite
(activiteID INTEGER PRIMARY KEY,
 ffaID INTEGER NOT NULL REFERENCES Client,
 releveID INTEGER NULL REFERENCES Releve,
 montant INTEGER NOT NULL,
 dateActivite DATE NOT NULL);
```

- clefs étrangères simples
- pas de redondance directe... quelle redondance reste-t-il?

16

Surcharge de clefs étrangères

```
Address(aid INTEGER PRIMARY KEY, addr TEXT);
Person(pid INTEGER PRIMARY KEY, name TEXT);
Comp(cid INTEGER PRIMARY KEY, info TEXT);
AddressLink
(aid INTEGER NOT NULL REFERENCES Address,
xid INTEGER, -- reference Person ou Comp !
is_xid_person BOOLEAN);
```

- schéma non relationnel : plus d'algèbre simple !
- doit être interprété par l'application...

17

solution tables liens indépendantes pour chaque table

```
AddressPersonLink
(aid INTEGER NOT NULL REFERENCES Address,
pid INTEGER NOT NULL REFERENCES Person);
AddressCompLink
(aid INTEGER NOT NULL REFERENCES Address,
cid INTEGER NOT NULL REFERENCES Comp);
```

solution une seule table lien (informations attachées au lien...)

```
AddressMixLink
(aid INTEGER NOT NULL REFERENCES Address,
pid INTEGER NULL REFERENCES Person,
cid INTEGER NULL REFERENCES Comp,
CHECK((pid IS NULL AND cid IS NOT NULL) OR
(cid IS NULL AND pid IS NOT NULL)));
```

18

Sous modélisation/traduction/typage dans une application

- types utilisés : DATE DATETIME *CHAR* *INT*
- aucune contrainte sur les valeurs, défauts non homogènes
- aucune contrainte d'intégrité référentielle
- certaines tables sans clefs primaires...

```
'INTERNSHIP_OUT' tinyint(1) default '0',
'AE' tinyint(1) default '0',
'S1' tinyint(1) default NULL,
'MARRIED' tinyint(1) default NULL,
'TITULAR' tinyint(1) default NULL,
'SNCF1225' tinyint(1) default NULL,
'FAKE' tinyint(1) NOT NULL default '0',
'DELETED' binary(1) NOT NULL default 'E',
'HISTORY_ACCESS' binary(1) NOT NULL default '\0',
'HISTORY_CHANGE' binary(1) NOT NULL default '\0',
-- quelques dates
'ID_DATE' date default '0000-00-00',
'SNCF1225_VALID' date default NULL,
-- clef primaire/étrangère (?)
'CODE' varchar(20) NOT NULL default '',
'CODE' int(11) NOT NULL,
'EVENT' int(11) NOT NULL default '0',
```

19

Contraintes insuffisantes

numéro de téléphone		
941.80.40	+1.412.268.75.53	(0483) 571281
47.76.34.37	+1.713.527-6077	45.81.74.61.01.45.81.77.77
98.49.44.44.	+31.15.78.24.19	INTERNATIONAL +44.1-703.592418
01.64.32.03.00	+39.6.68.60.12.32	READING
04.66.78.50.00	+43.1.310.56.08/71	CAMBRIDGE
(16).83.91.20.00	+44.1.253.4399	AGADIR
(1).46.83.64.64	+46.13281484 (282297)	00.212.227017.00.212.227125
+1.201-932-3699	+49.228.737..	Phone/Fax : (+86 10) 82 33 82 58 / 82 32 80 36
	04.72.72.80.00 ou nabr	01 34 03 00 00 ou NUMERO VERT : 0800 010 786 option 2

- attribut *tel_direct* numéro de téléphone, table *ecole*
- multi numéros, séparateurs . - () / , espaces, chaînes

```
CHECK (tel IS NULL OR tel SIMILAR TO '[+0-9.]+' )
```

20

Contraintes insuffisantes (suite)

```
email
screenmail address : ibmmail(XXXibmx9p)
www2.univ-reunion.fr/pocetelet/index
XXXnme@masi.ibp.fr ou XXXXar@masi.ibp.fr
XXXXen@ssf.uuucp ou sffXXXen (ancien mail)
XXX@inesc.uuucp eurokom : XXXe X. XXXdo inesc
XXXel.XXXch@free.fr & XXXel.XXXch@mb-ii.com
XXXond@diamant.unice.fr;XXXond@sophia.inria.fr
XXXXar@club-internet.fr;XXX.XXXXar@ideliance.com
internet : XXX@sco.com;uuucp : ...luunet;tsoccan;XXX
XXXre@imag.fr;XXXre@imag.uuunet.uu.net;imag;DOXXre
XXX.XXXrez@lip.ens-lyon.fr;XXXrez@lip.ens-lyon.fr
XXXin@icps.u-strasbg.fr; http://icps.u-strasbg.fr
XXXol@irisa.fr;usenet : ...Imovax;linria;irisa;DOXXol
XXXne@eik.ii.uib.no - XXXne@sunny.systems.cs.yale.edu
XXX@cornell.tn.cornell.edu;bitnet : XXX@cornell;bitnet
email : XXXXoux@mediapps.com;personal email : XXXXoux@wanadoo.fr
XXXXori@cc.ensmp.fr - XXXXori@cri.ensmp.fr - XXXX@urante.enpc.fr
```

- adresse internet (uunet, arpanet, usenet, bitnet, eunet...)

```
CHECK (email SIMILAR TO '[_a-z0-9.]+'@['_a-z0-9.]+' )
```

21

Où vérifier les contraintes ?

client interfaces (JS, Python, Ruby, PHP, Java)...

- liste choix cablée ? extraction base ?
- programmation plus riche ?
- plus commode pour gérer les erreurs
- client doit soumettre des modifications valides

serveur base de donnée

- toujours vérifié si interfaces multiples
- transactions complexes concurrentes
- un peu plus lent ? pas sûr
- expressions régulières très commodes

Faire les deux !

22

3 interfaces à Corrector

- web** élèves (ok) et profs (partiel, lent)
- scripts** conversion L^AT_EX vers SQL d'un exercices
- sql** mises à jours du schéma, corrections

Application scolarité

- une seule interface JS/PHP, vérification à ce niveau
- empêche la réutilisation des données : application fermée même si à base de logiciels libres : LAMP

23

Conclusion

- bonne compréhension de ce qui est modélisé
- subtilités...
- erreur difficilement rattrapables
- E/A simple, mais quelques inconvénients
 - pas d'unicité de la modélisation
 - pas de type pour les données
 - pas de clefs, primaires ou non
 - pas de contraintes explicites, pas toutes traduisibles
 - pas d'opérations prévues

24

List of Slides

- 1 Traduction du modèle E/A en relations
- 2 Modélisation entité-association (E/A)
- 3 Traductions des concepts
- 7 Conseils
- 9 Traduire le schéma Musique
- 10 Erreur type clef étrangère
- 11 Propagation d'identité : le triangle maudit
- 12 Traduction initiale en relations
- 14 Conséquences
- 15 Clefs primaires simples : la bonne solution
- 17 Surcharge de clefs étrangères
- 19 Sous modélisation/traduction/typage dans une application
- 20 Contraintes insuffisantes
- 21 Contraintes insuffisantes (suite)
- 22 Où vérifier les contraintes ?
- 23 3 interfaces à Corrector
- 23 Application scolarité
- 24 Conclusion