# Limites des structures plates

# Systèmes de Gestion de Base de Données (SGBD)

# Le modèle relationnel

Application III-c

Les affirmations suivantes sont-elles vraies ou fausses ? Justifier la réponse.

On se place dans une base de données nommée « bddPerso ».

* 1. Une ligne de table de « bddPerso » est appelée une entrée.
  2. Une table de « bddPerso » est définie par la structure relationnelle suivante :

Clients (idClient, nom, prénom)

Alors « nom » est un attribut de la tables « Clients ».

* 1. Dans la table « clients » de la question précédente, on peut prendre « nom » comme clé primaire.
  2. On définit une autre table de la manière suivante :

Connexions (idCommande, idClient, date) où « idClient » fait référence à la clé du même nom dans la table « clients ».

Alors « idClient » est alors une clé étrangère de la table « connexions ».

Deux tables modélisent la flotte de voitures d’un réseau de location de voitures.

|  |  |  |
| --- | --- | --- |
| **agences** | | |
| id\_agence | ville | département |
| 1 | Paris | 75 |
| 2 | Lyon | 69 |
| 3 | Marseille | 13 |
| 4 | Aubagne | 13 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **voitures** | | | | | |
| id\_voiture | marque | modèle | kilométrage | couleur | id\_agence |
| 1 | Renault | Clio | 12000 | Rouge | 2 |
| 2 | Peugeot | 205 | 22000 | Noir | 3 |
| 3 | Toyota | Yaris | 33000 | Noir | 3 |

Combien d’attributs possède la table « voitures » ? Quel est son degré ?

Quel est le domaine de l’attribut « id\_agence » dans la table « voitures » ?

Quel est le domaine de l’attribut « département » dans la table « voitures » ?

Quels sont les clés primaires/étrangères de ces deux tables ?

Application III-d

Schématiser dans un premier temps la base de données précédentes dans la cadre suivant puis le faire sur le site internet proposé. En plus des deux tables précédentes, vous devez créer une table « clients » qui regroupe les clients de l’agence et leur location. A vous de choisir les attributs et clés de cette table.

Application III-e

Une sandwicherie effectuant des livraisons à domicile dispose d’une base de données dont certains extraits de tables sont reproduits ici.

|  |  |
| --- | --- |
| **sandwich** | |
| nom\_sandwich | prix |
| Cheeseburger | 3.90 |
| Double cheese | 4.90 |
| Italien | 4.90 |
| Fois gras | 15.00 |

|  |  |  |  |
| --- | --- | --- | --- |
| **clients** | | | |
| nom | prénom | adresse | numéro\_client |
| Bernard | Alain |  |  |
| Bernard | Yves |  |  |

# Conception d’une base de données

1. **Exemple d’anomalie : redondance de données**

Lorsque les bases de données n’ont pas été suffisamment normalisées ou lorsqu’elles n’ont pas été conçues de sorte que les contraintes d’intégrité soient respectées après modification des relations, des *anomalies* peuvent apparaître, comme la *redondance de données*.

Considérons, par exemple, une base de données modélisant les produits achetés par une entreprise avec le schéma relationnel :

S = ((Produit, chaîne de caractère), (Fournisseur, chaîne de caractère), (Adresse, chaîne de caractère))

Un exemple de table pour ce schéma :

|  |  |  |
| --- | --- | --- |
| S | | |
| Produit | Fournisseur | Adresse |
| Tomate | Verger du sud | MIN Cavaillon |
| Ananas | Verger du sud | MIN Cavaillon |
| Origan | SA Epices | MIN Rungis |

L’adresse est redondante dans cette table, celle des Vergers du Sud apparaissant plusieurs fois, ce qui a un coût d’espace et conduit à des performances moindres de la base de données.

La solution typique consiste à séparer cette table en deux :

Fabricants = ((Fournisseur, chaîne de caractère), (Adresse, chaîne de caractère))

Produits = ((Produit, chaîne de caractère), (Fournisseur, chaîne de caractère))

|  |  |
| --- | --- |
| Fabricants | |
| Fournisseur | Adresse |
| Verger du sud | MIN Cavaillon |
| SA Epices | MIN Rungis |

|  |  |
| --- | --- |
| Produits | |
| Produit | Fournisseur |
| Tomate | Verger du sud |
| Ananas | Verger du sud |
| Origan | SA Epices |

Néanmoins cette solution n’est pas pleinement satisfaisante. Si la société Verger du sud change de nom ou est supprimée de la table « Fabricants », la table « Produits » aura une anomalie de mise à jour (ou de suppression), car les produits Tomate et Ananas ne seront plus référencés. Et si une nouvelle société prend le même nom qu’une déjà existante on aura une anomalie d’insertion.

1. **Elaboration d’un modèle conceptuel des données**

La première étape pour aboutir à un modèle permettant de stocker les données dans une base consiste à identifier les objets à représenter ainsi que leurs liens. Cela aboutit à un *modèle conceptuel des données (MCD)*, comme le *modèle entité-association* présenté ici.

**Définition n°7**

Une entité est une unité de base du système modélisé. Il peut s’agir d’individus, d’objets qui ont des propriétés communes.

**Définition n°8**

Une association représente un lien entre des entités. Elle peut être binaire et représenter un lien entre deux entités A et B, nommée par un verbe. Si chaque occurrence de l’entité A est liée à au plus une occurrence de l’entité B on parle de *relation binaire fonctionnelle*, sinon on parle *d’association non fonctionnelle.*

**Exemple :**

Prenons comme exemple l’étude simplifiée de la gestion d’un parc immobilier. Des propriétaires possèdent, éventuellement collectivement, un ou plusieurs appartements, un locataire occupant un appartement.

Trois entités se dégagent ici : les propriétaires, les locataires et les appartements. Il y a un lien représenté par le verbe « habiter » entre l’entité locataire et l’entité appartement. Ce lien est une relation binaire fonctionnelle dans la mesure où on admet qu’un locataire n’occupe qu’un seul appartement. Il y a un lien représenté par le verbe « posséder » entre l’entité propriétaire et l’entité appartement. Cette association n’est pas fonctionnelle car un propriétaire peut posséder plusieurs appartements et, réciproquement, ils peuvent co-propriétaires.

1. **D’un MCD aux tables**

D’un MCD, on en déduit des tables liées entre elles en utilisant les règles suivantes :

* Toute entité donne lieu à une table dont la clé primaire est l’identifiant de l’entité.
* Toute association binaire fonctionnelle est implémentée par la présence d’une clé étrangère dans la table qui représente l’entité origine de la dépendance fonctionnelle.
* Toute association non fonctionnelle génère une table dont la clé primaire est l’ensemble des clés primaires des tables qu’elle relie.

**Exemple :**

Dans l’exemple de la gestion immobilière, on a trois entités évidentes donc trois tables : « propriétaires », « locataires » et « appartements ». Pour chaque table, on crée une clé primaire qui correspond à l’identifiant de l’entité et on ajoute les attributs utiles.

|  |
| --- |
| **appartements** |
| \*id\_appart |
| adresse |

|  |
| --- |
| **propriétaires** |
| \*id\_proprio |
| nom  prénom |

|  |
| --- |
| **locataires** |
| \*id\_locataire |
| nom  prénom |

Dans le MCD, il y avait une relation binaire fonctionnelle entre « locataires » et « appartements ». On ajoute donc dans chacune de ces tables une clé étrangère faisant référence à l’autre table :

|  |
| --- |
| **appartements** |
| \*id\_appart  #id\_locataire |
| adresse |

|  |
| --- |
| **locataires** |
| \*id\_locataire  #id\_appartement |
| nom  prénom |

|  |
| --- |
| **propriétaires** |
| \*id\_proprio |
| nom  prénom |

Il avait également une association non fonctionnelle entre « propriétaires » et « appartements » qui donne lieu à la création d’une nouvelle table « proprioAppart » contenant les clé primaires des tables « propriétaires » et « appartements » .

|  |
| --- |
| **appartements** |
| \*id\_appart  #id\_locataire |
| adresse |

|  |
| --- |
| **locataires** |
| \*id\_locataire  #id\_appartement |
| nom  prénom |

|  |
| --- |
| **propriétaires** |
| \*id\_proprio |
| nom  prénom |

|  |
| --- |
| proprioAppart |
| #id\_appart  #id\_proprio |