

La modélisation logique relationnelle

Table des matières

I - Cours	3
1. Introduction au modèle relationnel	3
1.1. Niveau logique	3
1.2. Définition du modèle relationnel	3
2. Les concepts fondamentaux du modèle relationnel : attributs, enregistrement, domaine	4
2.1. Domaine	4
2.2. Exercice	5
2.3. Attribut et enregistrement	5
2.4. Exemple : La relation "Vol"	6
3. Clés et clés étrangères dans le modèle relationnel :	6
3.1. Clé	6
3.2. Déterminer les clés	7
3.3. Clé primaire et clés candidates	7
3.4. Clé artificielle	8
3.5. Clé étrangère	9
3.6. Référence entre relations	10
3.7. Schéma relationnel	10
3.8. Exemple de schéma relationnel pour la géographie	12
4. Synthèse	13
4.1. Synthèse : Schéma relationnel	13
4.2. Bibliographie commentée sur le modèle relationnel	13
5. Définition formelle d'une relation	14
5.1. Produit cartésien	14
5.2. Relation	14
II - Exercices	15
1. Exercice	15
2. Exercice : Lab I-	15
Contenus annexes	17
Glossaire	19
Abréviations	20
Bibliographie	21
Index	22



Le modèle relationnel est aux fondements des *SGBDR**. Il a été, et continue d'être, le modèle théorique dominant pour la représentation logique des bases de données, même si le mouvement NoSQL propose des alternatives.

Le modèle relationnel permet de reformuler le modèle conceptuel dans un formalisme - le tableau - beaucoup plus proche de l'implémentation informatique, bien qu'encore indépendant d'une solution technologique particulière.

1. Introduction au modèle relationnel

Objectifs

Connaître les fondements théoriques du modèle relationnel.

1.1. Niveau logique

Le niveau logique est le lien entre le niveau conceptuel et l'implémentation effective de l'application. Le modèle conceptuel étant un modèle formel, le modèle logique a pour vocation d'être également un modèle formel, mais spécifiant non plus la réalité existante ou recherchée comme le modèle conceptuel, mais les données telles qu'elles vont exister dans l'application informatique.

Pour assumer cette fonction, le modèle relationnel s'est imposé en réaction aux insuffisances des modèles antérieurs, les modèles hiérarchique et réseau, et de part la puissance de ses fondements mathématiques.

Encore aujourd'hui dominant le modèle relationnel est un fondement indispensable à la conception de bases de données.



Rappel

Méthodologie générale de conception d'une base de données (cf. p.17)

1.2. Définition du modèle relationnel

Introduction

Le modèle relationnel a été introduit par *Codd**, en 1970 au laboratoire de recherche d'IBM de San José. Il s'agit d'un modèle simple et puissant à la base de la majorité des bases de données, encore aujourd'hui.

Les objectifs du modèle relationnel, formulés par Codd, sont les suivants :

- Assurer l'indépendance des applications et de la représentation interne des données
- Gérer les problèmes de cohérence et de redondance des données
- Utiliser des langages de données basés sur des théories solides

Modèle relationnel



Définition

On appelle modèle relationnel un ensemble de concepts permettant de formaliser logiquement la description d'articles de fichiers plats, indépendamment de la façon dont ils sont physiquement stockés dans une mémoire numérique.

Le modèle relationnel inclut des concepts pour la **description** de données, ainsi que des concepts pour la **manipulation** de données.

Représenter le monde en tables



Fondamental

Le modèle relationnel permet de représenter les données que l'on va gérer à l'aide d'un très petit nombre de concepts très simples :

- Les relations ou tables : des lignes et des colonnes
- Les domaines de valeurs : chaque case d'une table prend une unique valeur dans un domaine pré-défini
- Les clés : il existe des cases dont les valeurs doivent être uniques et non nulles
- Les clés étrangères : il existe des cases qui doivent prendre une valeur existante dans les cases d'une autre table

Extension du modèle relationnel



Complément

Le modèle relationnel est un standard, normalisé par l'ISO à travers son langage, le SQL. Il se veut néanmoins dès l'origine extensible, pour permettre de gérer des données plus complexes que les données tabulaires. Le modèle relationnel-objet est né de cette extension.

2. Les concepts fondamentaux du modèle relationnel : attributs, enregistrement, domaine

Objectifs

Connaître les fondements théoriques du modèle relationnel.

2.1. Domaine

Domaine



Définition

Ensemble, caractérisé par un nom, dans lequel des données peuvent prendre leurs valeurs.



Remarque

Un domaine peut-être défini en *intension** (c'est à dire en définissant les propriétés caractéristiques des valeurs du domaine, on parle aussi de compréhension) ou en *extension** (c'est à dire en énumérant toutes les valeurs du domaine)

Domaines définis en intension



Exemple

- Tous les entiers
- Les réels inférieur à 5
- Les booléen (vrai ou faux)

- Toutes les chaînes de 1 à 255 caractères
- Les valeurs monétaires, définie comme des décimaux avec deux chiffres après la virgule
- Les dates, définies comme des chaînes de 10 caractères comprenant des chiffres et des tirets selon le patron "00-00-0000"
- Les salaires, définis comme des valeurs monétaires compris entre 15.000 et 100.000

Domaines définis en extension



- Couleur : {Bleu, Vert, Rouge, Jaune, Blanc, Noir}
- SGBD : {Hiérarchique, Réseau, Relationnel, Objet, Relationnel-Objet}

2.2. Exercice

Indiquez quelle définition et quel exemple correspondent respectivement aux mots **intension** et **extension**.

{bleu, rouge, vert}

Énonciation exhaustive de l'ensemble des objets du domaine

Le domaine des couleurs

Explicitation d'un domaine par la description de ses caractéristiques (en vue de sa compréhension abstraite, générale).

Intension	Extension

2.3. Attribut et enregistrement

Attribut



On appelle attribut d'une relation, une colonne de cette relation. Un attribut est caractérisé par un nom et un domaine dans lequel il prend ses valeurs.

Synonymes : Champs, Propriété, Colonne

Enregistrement



On appelle enregistrement d'une relation, une ligne de cette relation. Un enregistrement prend une valeur pour chaque attribut de la relation.

Synonymes : Tuple, N-uplet, Vecteur, Ligne

? Exemple

A	B
1	1
1	2
2	2

Relation R

La relation R comporte les deux attributs A et B et les trois enregistrements $\langle 1,1 \rangle$, $\langle 1,2 \rangle$ et $\langle 2,2 \rangle$

Attribut, domaine, ordre

🔍 Remarque

Un attribut se distingue d'un domaine car il peut ne comporter que certaines valeurs de ce domaine. Les colonnes de la relation ne sont pas ordonnées et elles ne sont donc repérées que par le nom de l'attribut.

Valeur nulle

🔍 Remarque

Un enregistrement peut ne pas avoir de valeur pour certains attributs de la relation, parce que cette valeur est inconnue ou inapplicable, sa valeur est alors "null".

2.4. Exemple : La relation "Vol"

? Exemple

Numero	Compagnie	Avion	Départ	Arrivée	Date
AF3245	Air France	747	Paris	Oulan Bator	01-08-2002
AF6767	Air France	A320	Paris	Toulouse	30-07-2002
KLM234	KML	727	Paris	Amsterdam	31-07-2002

Relation Vol

3. Clés et clés étrangères dans le modèle relationnel :

Objectifs

Connaître les notions de clés candidates, naturelles, artificielles, primaire, étrangère

Aborder le principe d'éclatement des relations et de non-redondance.

3.1. Clé

Clé

🔑 Définition

Une clé est un groupe d'attributs minimum qui permet d'identifier de façon univoque un tuple dans une relation.

💡 Fondamental

Toute relation doit comporter au moins une clé, ce qui implique qu'une relation ne peut pas contenir deux tuples identiques.

Attributs de clés unique et non null



Attention

Afin d'être déterminants pour l'identification d'un enregistrement, tous les attributs d'une clé doivent être valués, c'est-à-dire qu'aucun ne peut avoir de valeur *null*.

Dire qu'un groupe d'attributs est une clé implique qu'il est unique et non *null*.

Numéro d'étudiant



Exemple

- Le numéro d'étudiant d'une relation *Etudiant* est une bonne clé car il y aura systématiquement une valeur non nulle.
- Le groupe d'attributs (nom, prénom) d'une relation *Etudiant* est en général une mauvaise clé, car les homonymes existent.

3.2. Déterminer les clés

Détermination d'une clé

Définir un groupe d'attributs comme étant une clé nécessite une réflexion sémantique sur les données composant ces attributs, afin de s'assurer de leur unicité.



Fondamental

La définition des clés est un acte de **modélisation**, elle ne renvoie pas donc pas à une vérité intangible, mais à la réalité telle qu'elle est représentée dans le modèle que l'on élabore.



Exemple

L'attribut numéro de sécurité sociale d'une relation *personne* peut paraître une bonne clé a priori car son unicité est assurée. Mais tout le monde n'en dispose pas forcément (les enfants, des étrangers), donc ce n'est une clé que si l'on considère des personnes affiliées à la sécurité sociale.

3.3. Clé primaire et clés candidates

Clé primaire



Définition

Si plusieurs clés existent dans une relation, on en choisit une parmi celles-ci. Cette clé est appelée **clé primaire**.

La clé primaire est généralement choisie :

- de façon à être **immuable***, c'est à dire à ne jamais changer une fois valuée pour la première fois
- de façon à ce qu'elle soit la plus **simple**, c'est à dire portant sur le moins d'attributs et sur les attributs de domaine les plus basiques (entiers ou chaînes courtes typiquement).

Clés candidates



Définition

On appelle **clés candidates** l'ensemble des clés d'une relation qui n'ont pas été choisies comme clé primaire (elles étaient candidates à cette fonction).

3.4. Clé artificielle



Fondamental

S'il est impossible de trouver une clé primaire, ou que les clés candidates ne conviennent pas (non immuables, trop complexes...), il est possible de faire appel à une **clé artificielle**.

Clé artificielle



Définition

Une clé artificielle est un attribut supplémentaire ajouté au schéma de la relation, qui n'est lié à aucune signification, et qui sert uniquement à identifier de façon unique les enregistrements et/ou à simplifier les références de clés étrangères à l'intérieur de la base de données.

Clé signifiante



Définition

Une clé est signifiante si elle n'est pas artificielle.

Synonyme : Clé naturelle

Clé artificielle et niveau logique



Attention

Au niveau du modèle logique, il faut éviter la simplicité consistant à identifier toutes les relations avec des clés artificielles, et ne réserver cet usage qu'aux cas particuliers.



Conseil

1. Si au moins une clé naturelle *immuable** composée d'un seul attribut existe en choisir une parmi celles-ci comme clé primaire
2. Sinon, choisir une clé naturelle immuable composée de plusieurs attributs si elle ne pose pas de problème identifié
3. Toujours justifier l'emploi d'une clé artificielle (au niveau logique uniquement pour des raisons de complexité du modèle, les questions de performance sont étudiées au niveau physique)

Clé artificielle et niveau physique, évolutivité, maintenance et performance



Remarque

Au niveau de l'implémentation physique par contre, il est courant que des clés artificielles soient utilisées de façon systématique.

- Du point de vue de **l'évolutivité** de la BD, il existe toujours un risque qu'une clé non-artificielle perde sa propriété d'unicité ou de non-nullité.
- Du point de vue de **la maintenance** de la BD, il existe toujours un risque qu'une clé non-artificielle voit sa valeur modifiée (remise en cause de l'immutabilité) et dans ce cas, la répercussion de ce changement pour mettre à jour toutes les références peut poser problème.
- Du point de vue de **la performance** de la BD, les clés non-artificielles ne sont pas en général optimisées en terme de type et de taille, et donc peuvent limiter les performances dans le cadre des jointures. Précisons néanmoins qu'inversement les clés artificielles ont pour conséquence de systématiser des jointures qui auraient pu être évitées avec des clés primaires signifiantes.

Problème d'évolutivité posé par une clé signifiante



Soit le numéro de sécurité sociale la clé primaire d'une table d'une BD française, elle ne permettra pas d'entrer un individu non-français issu d'un pays ne disposant pas d'un tel numéro.

Problème de maintenance posé par une clé signifiante



Soit le numéro de sécurité sociale la clé primaire d'une table d'une BD centrale dont les données sont exploitées par d'autres tables d'autres BD qui viennent "piocher" dans cette BD pour leurs propres usages, sans que la BD centrale ne connaisse ses "clients". Soit une erreur dans la saisie d'un numéro de sécurité sociale dans la BD centrale, si ce numéro est corrigé, il faudrait (ce qui n'est pas possible dans notre cas) impérativement en avertir toutes les bases utilisatrices pour qu'elles mettent à jour leurs références.

Problème de performance posé par une clé signifiante



Soit le numéro de sécurité sociale la clé primaire d'une table comptant un million d'enregistrements, ce numéro est généralement un nombre à 13 chiffres ou une chaîne à 13 caractères, ce qui dans les deux cas est supérieur au nombre à 7 chiffres suffisant pour identifier tous les individus de la BD. Les performances seront donc toujours moins bonnes, lors des jointures, si une clé prend deux fois plus de place en mémoire que son optimum. Mais ajoutons que cette perte de performance n'a pas toujours de conséquence sur la réalité perceptible par les utilisateurs de la BD.

Inversement, soit une clé artificielle la clé primaire d'une table T1, par ailleurs référencée par une autre table T2. Soit le numéro de sécurité sociale un attribut clé de T1. Si l'on veut par requête disposer des informations de T2 ainsi que du numéro de sécurité sociale de T1, alors il faudra faire une jointure, tandis que si ce numéro signifiant avait été choisi comme clé primaire, cela n'aurait pas été nécessaire.

3.5. Clé étrangère

Clé étrangère



Une clé étrangère est un attribut ou un groupe d'attributs d'une relation R1 devant apparaître comme clé primaire dans une relation R2 afin de matérialiser une référence entre les tuples de R1 et les tuples de R2.

Une clé étrangère d'un tuple référence une clé primaire d'un autre tuple.



Seule une clé primaire peut être référencée par une clé étrangère.

C'est même la seule fonction de la clé primaire : être la clé qui peut être référencée par les clés étrangères.

Contrainte d'intégrité référentielle



Une clé étrangère respecte la contrainte d'intégrité référentielle si sa valeur est effectivement existante dans la clé primaire d'un tuple de la relation référencée, ou si sa valeur est *null*.

Une clé étrangère qui ne respecte pas la contrainte d'intégrité référentielle exprime un lien vers un tuple qui n'existe pas et donc n'est pas cohérente.

3.6. Référence entre relations

Le modèle relationnel a pour objectif la structuration de données selon des relations. L'enjeu est de parvenir à traduire un modèle conceptuel en modèle logique relationnel. Or, il n'y a pas de notion d'association en relationnel, donc il faudra pouvoir traduire les associations avec les concepts dont on dispose : relation, clé, clé étrangère.

Afin de représenter des références entre relations dans un modèle relationnel, la seule solution est de stocker l'information dans une relation, et donc que certains attributs d'une relation servent à pointer sur d'autres relations.



Attention

Il n'y a pas vraiment de référence ou de lien en relationnel, puisque nous ne disposons que de tables, de clés, de clés étrangères et de valeurs.

On va donc devoir se servir de ces outils pour matérialiser une notion de référence.

Référence



Méthode

La référence entre deux tuples $T1$ et $T2$ de deux relations différentes est exprimable par une valeur identique entre une clé étrangère du tuple $T1$ et la clé primaire de l'autre tuple $T2$.

Synonyme : Lien



Exemple

R1		R2	
#a1	a2=>R2	#b1	b2
A	2	1	F
B	1	2	F
C	1	3	F
D	4	4	V

L'attribut $a2$ de la relation $R1$ référence l'attribut $b1$ de la relation $R2$ car $a2$ est une clé étrangère de $R1$ vers $R2$ ($b1$ est la clé primaire de $R2$).

Ici on a donc par exemple les tuples identifiés par B et C de $R1$ qui référencent le tuple identifié par 1 dans $R2$.

3.7. Schéma relationnel

Schéma d'une relation



Définition

Le schéma d'une relation définit cette relation en intension. Il est composé :

- du nom de la relation,
- de la liste de ses attributs avec les domaines respectifs dans lesquels ils prennent leurs valeurs,
- de la clé primaire,
- des clés étrangères,
- des clés candidates.

Schéma relationnel d'une base de donnée



Définition

Le schéma relationnel d'une BD est la définition en intension de cette BD (par opposition à l'instance de la BD qui est une extension de la BD). Il est composé de l'ensemble des schémas de chaque relation de la BD.

Relation



Syntaxe

```
1 Relation (Attribut1:Domaine1, Attribut2:Domaine2, ... , AttributN:DomaineN)
```

La relation "Relation" contient N attributs chacun défini sur son domaine.

Clé primaire



Syntaxe

```
1 Relation (#Attribut1:Domaine1, ... , #AttributM:DomaineM, ... , AttributN:DomaineN)
```

La clé de la relation "Relation" est composée des attributs "Attribut1" à "AttributM" (attribut précédés de # ou bien soulignés)

En général on note la clé primaire en premier dans la relation.

Clé étrangère



Syntaxe

```
1 Relation1 (... , AttributM=>Relation2, ... , AttributN=>Relation2)
```

La relation "Relation1" comporte une clé étrangère (composée des attributs "AttributM" à "AttributN") référençant la clé primaire de "Relation2". Bien sûr il peut exister plusieurs clés étrangères vers plusieurs relations distinctes. Une clé étrangère et sa clé primaire référencée sont toujours composées du même nombre d'attributs. Il n'est pas nécessaire de préciser les domaines des attributs appartenant à la clé étrangère car ce sont forcément les mêmes que ceux de la clé primaire référencée. Il n'est pas non plus en général nécessaire de préciser dans le schéma relationnel quels attributs de la clé étrangère référencent quels attributs de la clé primaire (cela est généralement évident) mais il est possible de la faire en notant "Attribut=>Relation.Attribut".

En général on note les clés étrangères en dernier dans la relation, sauf pour les clés étrangères qui font partie de la clé primaire (clés identifiantes).

Clé candidates



Syntaxe

```
1 Relation1 (...AttributM:DomaineM, ..... ) avec AttributM clé
```

Les clés candidates doivent être notées sur le schéma relationnel :

- S'il n'y a qu'une ou deux clés candidates, les noter directement après la définition de la relation
- S'il y a beaucoup de clés, pour ne pas trop alourdir la notation, renvoyer à un tableau à part

Clés candidates et clé primaire



Attention

La notation $R(\#a, \#b)$ signifie toujours que R a comme clé primaire (a, b) , et non que R aurait deux clés a et b (dont on ne saurait pas laquelle est primaire).

La notation $R(\#a, b)$ avec b clé signifie bien que a et b sont deux clés de R , et que a est primaire.

Il ne faut pas confondre **une clé composée de deux attributs** avec **deux clés**.

3.8. Exemple de schéma relationnel pour la géographie

? Exemple

```

1 Personne (#Numero:Entier, Nom:Chaîne, Prénom:Chaîne, LieuNaissance=>Ville)
2 Pays (#Nom:Chaîne, Population:Entier, Superficie:Réel, Dirigeant=>Personne)
3 Région (#Pays=>Pays, #Nom:Chaîne, Superficie, Dirigeant=>Personne)
4 Ville (#CodePostal:CP, Nom:Chaîne, Pays=>Région.Pays, Région=>Région.Nom,
Dirigeant=>Personne)

```

? Exemple

Exemple d'instance de la base de données

Personne

#Numero	Nom	Prenom	LieuNaissance
1	Durand	Pierre	60200
2	Dupont	Marie	60200

Pays

#Nom	Population	Superficie	Dirigeant
France	60	500001,01	2
Allemagne	80	600000,23564	2
Espagne	40	350000,1	1

Région

#Pays	#Nom	Superficie	Dirigeant
France	Picardie	50	1
Espagne	Picardie	40	1
France	Normandie	30	2

Ville

#CodePostal	Nom	Pays	Région	Dirigeant
F60200	Compiègne	France	Picardie	1
F60300	Senlis	France	Picardie	2
F60301	Senlis	France	Picardie	2
E8000	Senlis	Espagne	Picardie	2

Le schéma relationnel précédent décrit :

- **Des personnes**

Elles sont identifiées par un numéro qui est en fait une clé artificielle. En effet, même une clé composée de tous les attributs (Nom, Prénom, LieuNaissance) laisse une possibilité de doublons (homonymes nés dans la même ville).

La clé étrangère LieuNaissance fait référence à la relation Ville, et plus précisément à sa clé primaire CodePostal, ce qui est laissé implicite car évident.

- **Des pays**

Ils sont identifiés par leur nom, puisque deux pays ne peuvent avoir le même nom.

Les pays sont dirigés par des personnes, et ce lien est matérialisé par la clé étrangère Dirigeant.

- **Des régions**

Elles font partie d'un pays et ont un nom. Deux régions de pays différents pouvant avoir le même nom, il faut utiliser une clé primaire composée à la fois du nom de la région et du nom du pays, qui est une clé étrangère (le nom est appelé clé locale car il n'est pas suffisant pour identifier un tuple de la relation Région, et la clé étrangère vers la relation Pays est appelée clé identifiante).

- **Des villes**

Elles sont identifiées par un code postal qui est unique dans le monde (en utilisant le préfixe de pays de type "F-60200"). Ce code postal a pour domaine CP qui est une chaîne composée d'une ou deux lettres, d'un tiret, puis d'une série de chiffres.

Le lien d'appartenance entre une ville et une région est matérialisé par la clé étrangère composée des deux attributs Pays et Région. Cette clé référence la clé primaire de la relation Région, également composée de deux attributs. Pour clairement expliciter les références (bien que sémantiquement la dénomination des attributs ne laisse pas de place au doute) on utilise la syntaxe Région.Pays et Région.Nom.

4. Synthèse

4.1. Synthèse : Schéma relationnel

Schéma relationnel

Un schéma relationnel permet une formalisation d'un modèle logique.

- Relation ou table

Sous-ensemble d'un produit cartésien

- Attribut ou colonne

Prend ses valeurs dans un domaine

- Enregistrement ou ligne

Pose une valeur (y compris la valeur nulle) pour chaque attribut

- Clé

Groupe d'attributs ayant un rôle d'identification au sein d'un enregistrement

- Clé candidate

Identifie de façon unique un enregistrement

- Clé primaire

Clé candidate choisie pour représenter un enregistrement pour sa facilité d'usage

- Clé étrangère

Référence la clé primaire d'un tuple d'une autre relation pour exprimer un lien

4.2. Bibliographie commentée sur le modèle relationnel

Synthèses



*SQL2 SQL3, applications à Oracle **

Une définition synthétique et efficace du domaine relationnel : relation, domaine, attribut, clé, intégrité, opérateurs (Premier chapitre).

5. Définition formelle d'une relation

5.1. Produit cartésien

Produit cartésien



Définition

Le produit cartésien, noté "X", des domaines D_1, D_2, \dots, D_n , noté " $D_1 \times D_2 \times \dots \times D_n$ " est l'ensemble des tuples (ou n-uplets ou vecteurs) $\langle V_1, V_2, \dots, V_n \rangle$ tel que V_i est une valeur de D_i et tel que toutes les combinaisons de valeurs possibles sont exprimées.



Exemple

```
1 D1 = {A, B, C}
2 D2 = {1, 2, 3}
3 D1 X D2 = {<A,1>, <A,2>, <A,3>, <B,1>, <B,2>, <B,3>, <C,1>, <C,2>, <C,3>,}
```

5.2. Relation

Relation



Définition

Une relation sur les domaines D_1, D_2, \dots, D_n est un sous-ensemble du produit cartésien " $D_1 \times D_2 \times \dots \times D_n$ ". Une relation est caractérisée par un nom.

Synonymes : Table, tableau



Syntaxe

On peut représenter la relation R sur les domaines D_1, \dots, D_n par une table comportant une colonne pour chaque domaine et une ligne pour chaque tuple de la relation.

D_1	...	D_n
V_1	...	V_n
...
V_1	...	V_n

Relation R



Remarque

Une relation est définie en extension, par l'énumération des tuples la composant.

Exercices



1. Exercice

[10 min]

Question 1

Écrivez le résultat du produit cartésien $R1 \times R2$

R1

a	1
b	2

R2

x
NULL

Question 2

Écrivez le résultat du produit cartésien $R1 \times R2$

R1

x	1
---	---

R2

x
y

2. Exercice : Lab I-

Description du problème

[20 min]

Un laboratoire souhaite gérer les médicaments qu'il conçoit.

- Un médicament est décrit par un nom, qui permet de l'identifier. En effet il n'existe pas deux médicaments avec le même nom. Un médicament comporte une description courte en français, ainsi qu'une description longue en latin. On gère aussi le conditionnement du médicament, c'est à dire le nombre de pilules par boîte (qui est un nombre entier).

- À chaque médicament on associe une liste de contre-indications, généralement plusieurs, parfois aucune. Une contre-indication comporte un code unique qui l'identifie, ainsi qu'une description. Une contre-indication est toujours associée à un et un seul médicament.

Exemple de données

Afin de matérialiser notre base de données, nous obtenons les descriptions suivantes :

- Le **Chourix** a pour description courte « *Médicament contre la chute des choux* » et pour description longue « *Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae, sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare.* ». Il est conditionné en boîte de 13.

Ses contre-indications sont :

- CI1 : Ne jamais prendre après minuit.
- CI2 : Ne jamais mettre en contact avec de l'eau.

- Le **Tropas** a pour description courte « *Médicament contre les dysfonctionnements intellectuels* » et pour description longue « *Suspendisse lectus leo, consectetur in tempor sit amet, placerat quis neque. Etiam luctus porttitor lorem, sed suscipit est rutrum non.* ». Il est conditionné en boîte de 42.

Ses contre-indications sont :

- CI3 : Garder à l'abri de la lumière du soleil

Question 1

Dessiner des relations instanciées (en extension donc) remplies avec les données fournies en exemple.

Question 2

Écrivez le schéma relationnel (définition en intension donc) permettant de représenter une base de données relationnelle pour le laboratoire.

Contenus annexes

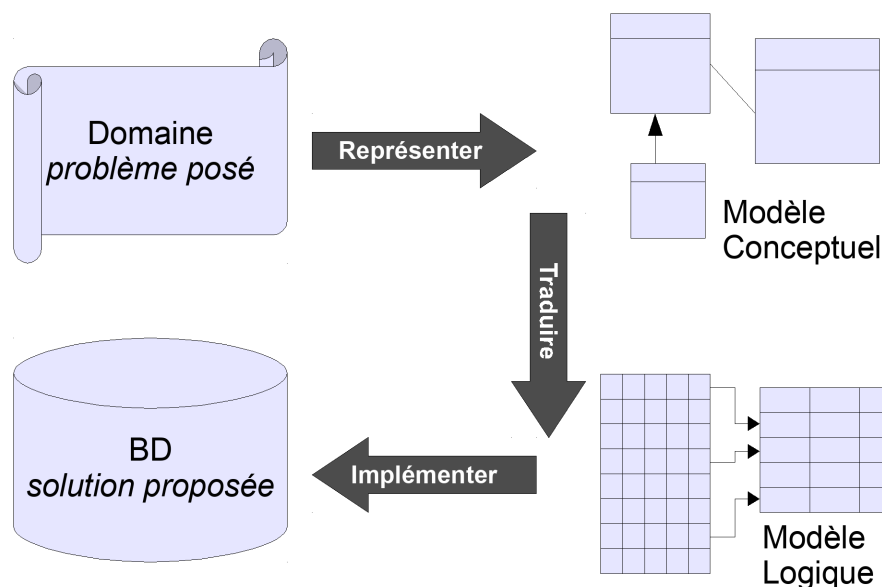


1. Méthodologie générale de conception d'une base de données

Étapes de la conception d'une base de données



1. **Analyse** de la situation existante et des besoins (clarification)
2. Création d'un **modèle conceptuel** qui permet de représenter tous les aspects importants du problème
3. Traduction du modèle conceptuel en **modèle logique** (et normalisation de ce modèle logique)
4. Implémentation d'une **base de données** dans un SGBD, à partir du modèle logique (et optimisation)



Processus de conception d'une base de données

On distingue quatre étapes dans la conception d'une base de données :

- **L'analyse**

Elle consiste à étudier le problème et à consigner dans un document, la note de clarification, les besoins, les choix, les contraintes.

- **La modélisation conceptuelle**

Elle permet de décrire le problème posé, de façon non-formelle (en générale graphique), en prenant des hypothèses de simplification. Ce n'est pas une description du réel, mais une représentation simplifiée d'une réalité.

- **La modélisation logique**

Elle permet de décrire une solution, en prenant une orientation informatique générale (type de SGBD typiquement), formelle, mais indépendamment de choix d'implémentation spécifiques.

- **L'implémentation**

Elle correspond aux choix techniques, en terme de SGBD choisi et à leur mise en œuvre (programmation, optimisation...).



Fondamental

- **Bien analyser** le problème posé en amont
- **Bien modéliser** le problème au niveau conceptuel avant de passer au niveau logique et à l'implémentation

L'importance de l'étape d'analyse



Conseil

La première étape de la conception repose sur l'analyse de l'existant et des besoins. De la qualité de la réalisation de cette première étape dépendra ensuite la pertinence de la base de données par rapports aux usages. Cette première étape est donc essentielle et doit être menée avec soins.

Si la première étape est fondamentale dans le processus de conception, elle est aussi la plus délicate. En effet, tandis que des formalismes puissants existent pour la modélisation conceptuelle puis pour la modélisation logique, la perception de l'existant et des besoins reste une étape qui repose essentiellement sur l'expertise d'analyse de l'ingénieur.

L'importance de l'étape de modélisation conceptuelle



Conseil

Étant donnée une analyse des besoins correctement réalisée, la seconde étape consiste à la traduire selon un modèle conceptuel. Le modèle conceptuel étant formel, il va permettre de passer d'une spécification en langage naturel, et donc soumise à interprétation, à une spécification non ambiguë. Le recours aux formalismes de modélisation tels que *E-A** ou *UML** est donc une aide fondamentale pour parvenir à une représentation qui ne sera plus liée à l'interprétation du lecteur.

La traduction d'un cahier des charges spécifiant l'existant et les besoins en modèle conceptuel reste néanmoins une étape délicate, qui va conditionner ensuite l'ensemble de l'implémentation informatique. En effet les étapes suivantes sont plus mécaniques, dans la mesure où un modèle logique est déduit de façon systématique du modèle conceptuel et que l'implémentation logicielle est également réalisée par traduction directe du modèle logique.

Les étapes de traduction logique et d'implémentation



Remarque

Des logiciels spécialisés sont capables à partir d'un modèle conceptuel d'appliquer des algorithmes de traduction qui permettent d'obtenir directement le modèle logique, puis les instructions pour la création de la base de données dans un langage orienté données tel que SQL. L'existence de tels algorithmes de traduction montre que les étapes de traduction logique et d'implémentation sont moins complexes que les précédentes, car plus systématiques.

Néanmoins ces étapes exigent tout de même des compétences techniques pour optimiser les modèles logiques (normalisation), puis les implémentations en fonction d'un contexte de mise en œuvre matériel, logiciel et humain.

Glossaire



Clé alternative (alternate key)

En relationnel les clés alternatives sont les clés candidates non retenues comme clés primaires.

Extension

L'extension est l'explicitation d'un domaine par l'énonciation exhaustive de l'ensemble des objets du domaine.

Elle s'oppose à l'intension qui est une description abstraite des caractéristiques du domaine.

- Exemple : {bleu, rouge, vert}
- Contre-exemple : Le domaine des couleurs

Immuable

Un attribut immuable est un attribut dont la valeur ne change plus jamais une fois fixée.

Synonymes : *frozen*, *immutable* en anglais (ou parfois en français comme anglicisme)

Intension

L'intension est l'explicitation d'un domaine par la description de ses caractéristiques (en vue de sa compréhension abstraite, générale).

Elle s'oppose à l'extension qui est l'énonciation exhaustive de l'ensemble des objets du domaine.

- Exemple : Le domaine des couleurs
- Contre-exemple : {bleu, rouge, vert}

Abréviations



E-A : Entité-Association

SGBDR : Système de Gestion de Bases de Données Relationnelles

UML : Unified Modeling Language

Bibliographie



Codd EF, *A relational model for large shared data banks*, Communications de l'ACM, juin 1970.

Delmal Pierre. *SQL2 SQL3, applications à Oracle*. De Boeck Université, 2001.

Index



Algèbre.....	14
Analyse	17
Association	10
Attribut	5, 6, 6, 8, 9
Clé.....	6, 7, 8, 9
Clé artificielle.....	8
Clé candidate.....	7
Clé primaire	7, 8
Clé signifiante	8
Codd.....	3
Conception	17
Conceptuel	17
Domaine.....	4, 14, 14
Enregistrement.....	5, 6
Lien	10
Logique	3, 3, 3, 4, 6
Modèle	3, 3, 4, 6, 10, 12
Modélisation.....	7
Opération.....	14
Produit.....	14, 14
Référence.....	10
Relation.....	5, 6, 6, 8, 9, 10, 10, 14
Relationnel. 3, 3, 3, 4, 6, 10, 12, 14,	14
Relationnel-objet.....	3, 3
Schéma	10, 12
Tuple.....	5