

Bases de Données : Modèle Logique (Algèbre Relationnelle)

Stéphane Devismes

Université Grenoble Alpes

26 août 2020

Plan

- 1 Introduction
- 2 Classes
- 3 Asso. multi-mono
- 4 Asso. multi-multi
- 5 Classes-Associations
- 6 Asso. mono-mono (rare)
- 7 Classes Faibles
- 8 Héritages

Forme intermédiaire

Traduction du modèle conceptuel (diagrammes) vers le modèle logique (algèbre relationnelle).

Traduction du modèle logique (algèbre relationnelle) vers le modèle physique (SQL).

Contraintes

Il y a des contraintes qui ne peuvent être exprimées que comme des commentaires dans chacun des 3 modèles (si on se limite à SQL pour le modèle physique).

Mais ce ne sont pas tout-à-fait les mêmes contraintes dans les 3 modèles...

À la fin, en utilisant les déclencheurs de PL/SQL on doit pouvoir exprimer toutes les contraintes.

On pourrait aussi gérer les contraintes au niveau de l'application (par exemple, JDBC), mais c'est déconseillé.

Clé primaire

La traduction du modèle conceptuel (diagrammes) vers le modèle logique (algèbre relationnelle) que nous présentons ici propose, **pour chaque relation** construite, un **ensemble d'attributs**

- dont les valeurs **déterminent** les valeurs de tous les attributs.

Dans l'immense majorité des cas,

- cet ensemble est **minimal** pour la propriété précédente.

Dans la suite, on le qualifiera donc (de manière un peu abusive) de **clé primaire**.

Dans le cas où la propriété de minimalité n'est pas satisfaite, il s'agit d'une **superclé**, c'est-à-dire un ensemble d'attribut A contenant (au moins) une clé. Il suffit donc d'**extraire une clé C de la superclé A** et d'utiliser celle-ci comme contrainte de clé primaire.

Projection : exemple

R_1		
A	B	C
a	b	c
a	y	c
x	y	z
u	v	w
r	s	t
r	s	z

$$R_1[A] = \{a, x, u, r\}$$

$$R_1[A, B] = \{\langle a, b \rangle, \langle a, y \rangle, \langle x, y \rangle, \langle u, v \rangle, \langle r, s \rangle\}$$

Projection

Pour tout relation R et tout ensemble A d'attributs de R , on note $R[A]$ la **projection de R sur A** .

Le résultat d'une projection est un **ensemble**.

Clés étrangères

Soient R_1 et R_2 deux relations, K_2 la clé primaire de R_2 , et supposons que les attributs qui composent K_2 sont aussi des attributs de R_1 , alors la **contrainte de clé étrangère** ou **contrainte d'intégrité référentielle** disant que les attributs K_2 de R_1 **font référence** à la clé primaire K_2 de R_2 est notée :

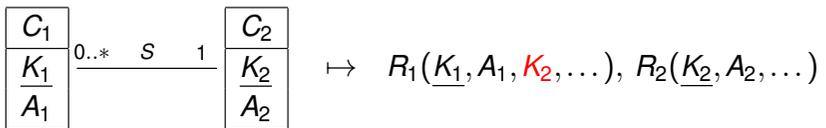
$$R_1[K_2] \subseteq R_2[K_2].$$

Fonction totale

$$I_1 = 0..* \text{ et } I_2 = 1$$

On doit ajouter la contrainte que les attributs K_2 dans R_1 ont toujours une valeur.

Lorsque B est un ensemble d'attributs de R , on note « B pas NULL dans R » pour dire que chacun des attributs de B a une valeur en chaque ligne de la table R .



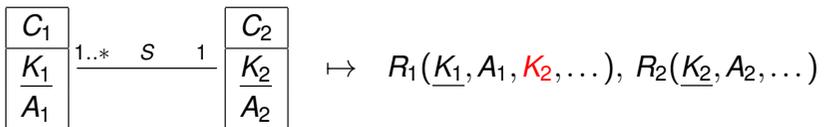
$$R_1[K_2] \subseteq R_2[K_2]$$

K_2 pas NULL dans R_1

Fonction totale surjective

$$I_1 = 1..* \text{ et } I_2 = 1$$

On a à la fois $R_1[K_2] = R_2[K_2]$ et chaque attribut de K_2 a une valeur dans R_1 .



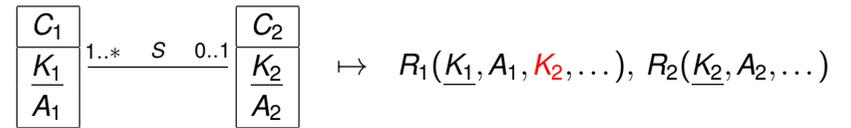
$$R_1[K_2] = R_2[K_2]$$

K_2 pas NULL dans R_1

Fonction partielle surjective

$$I_1 = 1..* \text{ et } I_2 = 0..1$$

On doit renforcer la contrainte $R_1(K_2) \subseteq R_2(K_2)$ en égalité $R_1[K_2] = R_2[K_2]$.



$$R_1[K_2] = R_2[K_2]$$

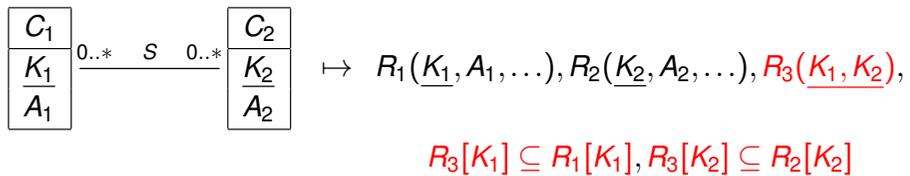
Exemple (1/2)



Exemple (2/2)



Cas de base : $l_1 = l_2 = 0..*$



Méthode

Soit S une association multi-multi entre C_1 et C_2 .

Soient R_1 et R_2 les relations construites à partir de C_1 et C_2 , et soient K_1 et K_2 leurs clés primaires.

Alors S est traduite par une relation R_3 .

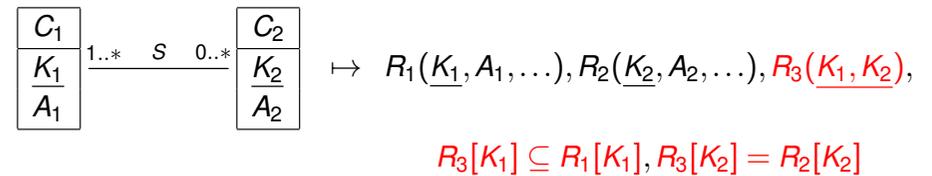
Les attributs de R_3 correspondent à l'ensemble des attributs de K_1 et de K_2 (avec renommage si besoin pour ne pas les confondre).

La clé primaire de R_3 est formée de tous ses attributs.

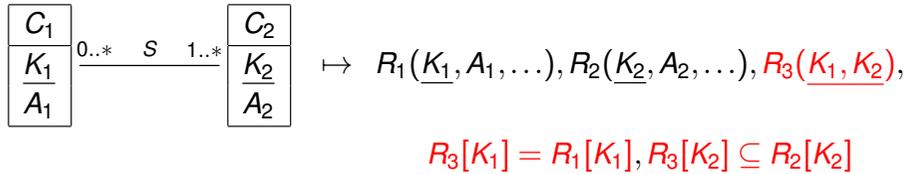
De plus on a deux contraintes de clé étrangère : les attributs K_1 de R_3 font référence à la clé de R_1 , et les attributs K_2 de R_3 font référence à la clé de R_2 :

$$R_3[K_1] \subseteq R_1[K_1] \text{ et } R_3[K_2] \subseteq R_2[K_2].$$

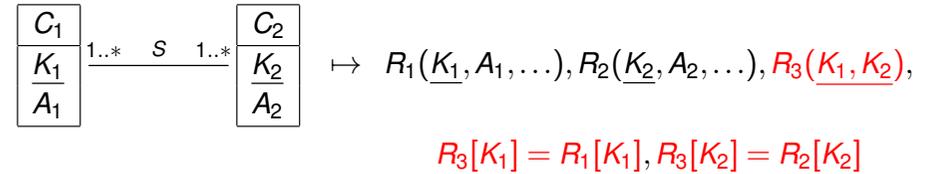
Cas de base : $l_1 = 1..*$ et $l_2 = 0..*$



Cas de base : $l_1 = 0..*$ et $l_2 = 1..*$



Cas de base : $l_1 = 1..*$ et $l_2 = 1..*$



Exemple

Association multi-multi « est écrite par »



Exemple récapitulatif

En résumé : 5 tables

- Auteurs (nom, prenom)
- Oeuvres (isbn, titre)
- Livres (numL, isbn, numA)
- Ecrivains (isbn, nom, prenom)
- Adhérents (numA, nom, prenom, adresse)

avec 4 contraintes de clés étrangères et 2 autres contraintes :

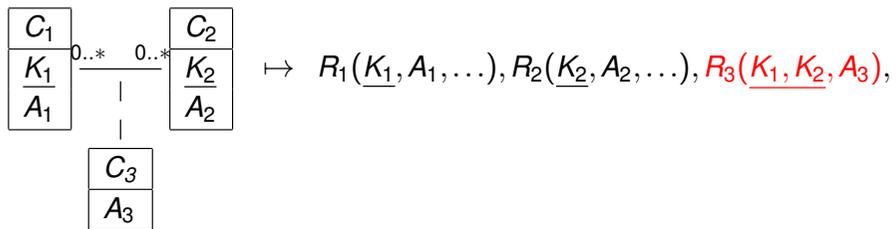
- Livres [numA] \subseteq Adhérents [numA]
- Livres [isbn] \subseteq Oeuvres [isbn]
- Ecrivains [isbn] \subseteq Oeuvres [isbn]
- Ecrivains [nom, prenom] \subseteq Auteurs [nom, prenom]
- isbn pas NULL dans Livres
- Un adhérent emprunte au plus 5 livres

Méthode pour le cas multi-multi

Soit C_3 une classe-association **multi-multi** entre C_1 et C_2 .

c'est similaire au cas de l'association **multi-multi**

Ici on traite le cas de base : $l_1 = l_2 = 0..*$



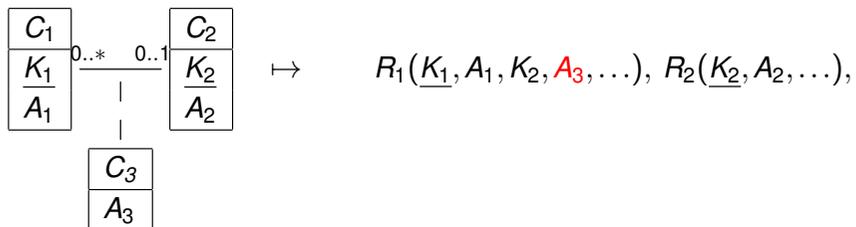
$$R_3[K_1] \subseteq R_1[K_1], R_3[K_2] \subseteq R_2[K_2]$$

Méthode pour le cas multi-mono (rare)

Soit C_3 une classe-association **multi-mono** entre C_1 et C_2 .

c'est similaire au cas de l'association **multi-mono**

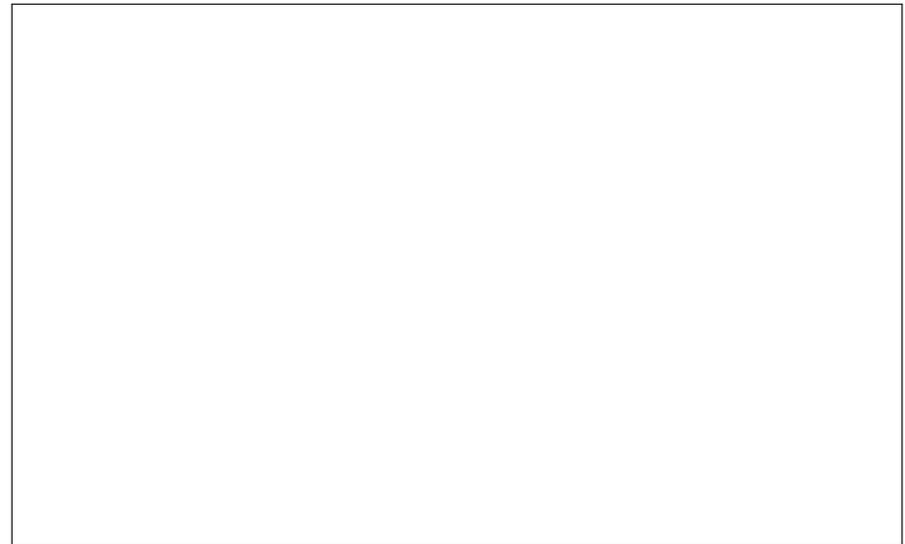
Ici on traite le cas de base : $l_1 = 0..*$ et $l_2 = 0..1$



$$R_1[K_2] \subseteq R_2[K_2]$$

K_2 est NULL dans $R_1 \Rightarrow A_3$ est NULL dans R_1

Exemple : Elèves / Diplôme



Exemple : ajouter la date d'emprunt

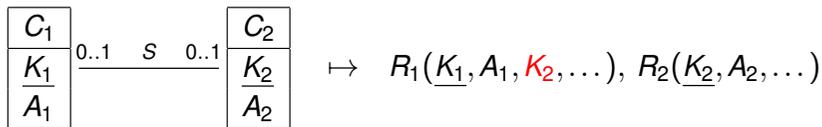


Méthode (1/3)

Soit S une association mono-mono de C_1 et C_2 . Soient R_1 et R_2 les relations construites à partir de C_1 et C_2 .

Cas : $I_1 = 0..1$ et $I_2 = 0..1$.

Alors S est traduite en ajoutant à R_1 les attributs de K_2 avec la contrainte de clé étrangère disant que ces attributs font référence à la clé primaire K_2 de R_2 et une contrainte d'unicité de K_2 dans R_1 . (Les rôles de R_1 et R_2 sont interchangeables.)



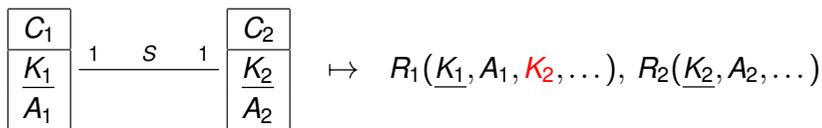
$R_1[K_2] \subseteq R_2[K_2]$
 K_2 est unique dans R_1

Méthode (3/3)

Soit S une association mono-mono de C_1 et C_2 . Soient R_1 et R_2 les relations construites à partir de C_1 et C_2 .

Cas : $I_1 = 1$ et $I_2 = 1$.

Alors S est traduite en ajoutant à R_1 les attributs de K_2 avec la contrainte de clé étrangère disant que ces attributs font référence à la clé primaire K_2 de R_2 . De plus K_2 est unique et non-nulle dans R_1 . (Les rôles de R_1 et R_2 sont interchangeables.)



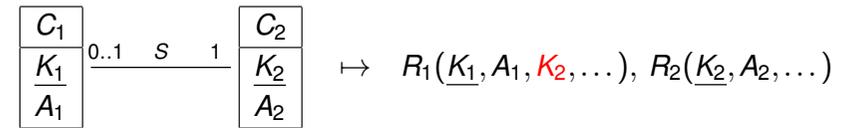
$R_1[K_2] = R_2[K_2]$
 K_2 est unique dans R_1
 K_2 est non-nulle dans R_1

Méthode (2/3)

Soit S une association mono-mono de C_1 et C_2 . Soient R_1 et R_2 les relations construites à partir de C_1 et C_2 .

Cas : $I_1 = 0..1$ et $I_2 = 1$.

Alors S est traduite en ajoutant à R_1 les attributs de K_2 avec la contrainte de clé étrangère disant que ces attributs font référence à la clé primaire K_2 de R_2 . De plus K_2 est unique et non-nulle dans R_1 .



$R_1[K_2] \subseteq R_2[K_2]$
 K_2 est unique dans R_1
 K_2 est non-nulle dans R_1

Exemples

Plusieurs exemples possibles :

- Employé/Ordinateur : chaque ordinateur est utilisé par exactement 1 employé et chaque employé travaille sur 0 ou 1 ordinateur
- Personne en couple (boucle sur une seule table)
- ...

Nous traitons maintenant le premier cas.

Exemple : Employé/Ordinateur



Méthode (1/2)

Soit $C_2 \Rightarrow C_1$ une référence de C_2 (classe faible) vers C_1 .

Soit R_1 la relation construite à partir de C_1 .

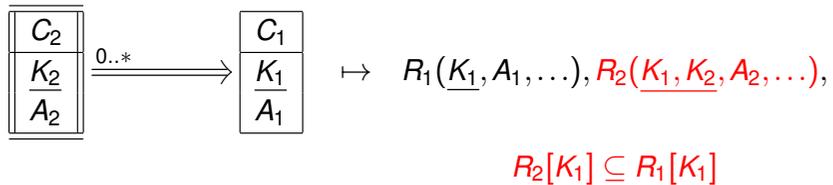
Alors C_2 est traduite par une relation R_2 .

Les attributs de R_2 sont formés de tous les attributs de K_1 auxquels on ajoute tous ceux de C_2 .

La clé primaire de R_2 est formée de K_1 et du complément de clé primaire de C_2 .

Méthode (2/2)

- En général :



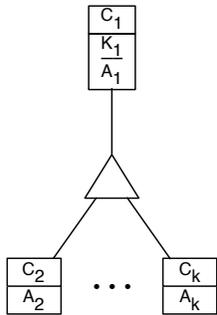
- Si $l_2 = 1..*$ alors la contrainte est $R_2[K_1] = R_1[K_1]$.

Classe faible : exemple

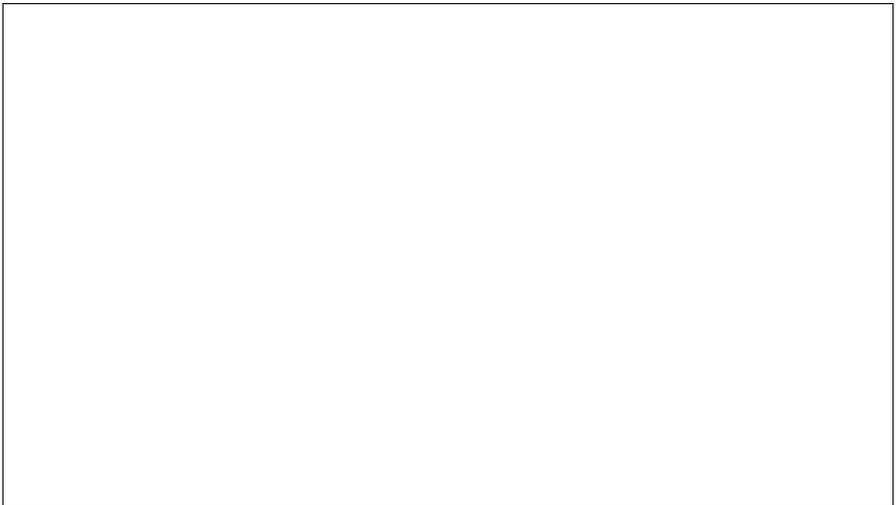


3 Méthodes

Nous considérons l'héritage suivant : les classes C_2, \dots, C_k ($k \geq 2$) sont des **sous-classes** de C_1 .



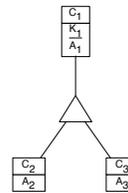
Exemple



Première Méthode

- Une relation R_1 pour C_1 .
 - Les attributs de R_1 sont ceux de C_1 .
 - La clé primaire de R_1 est la clé K_1 de C_1 .
- Une relation R_i pour toute classe C_i avec $2 \leq i \leq k$.
 - Les attributs de R_i sont les attributs de K_1 + ceux de C_i .
 - La clé primaire de R_i est K_1 .
 - On a une contrainte de clé étrangère pour chaque R_i avec $2 \leq i \leq k$.
 - Pour tout i, j avec $2 \leq i < j \leq k$ on a une contrainte $R_i[K_1] \cap R_j[K_1] = \emptyset$.
 - Si l'héritage est complet, on en plus la contrainte $R_1[K_1] = \bigcup_{i=2}^k R_i[K_1]$.

Exemple pour $k = 3$.



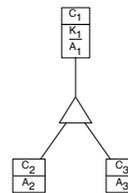
$$\begin{aligned}
 &R_1(K_1, A_1, \dots), R_2(K_1, A_2), R_3(K_1, A_3), \\
 &R_2[K_1] \subseteq R_1[K_1] \\
 &R_3[K_1] \subseteq R_1[K_1] \\
 &R_2[K_1] \cap R_3[K_1] = \emptyset \\
 &\text{Si l'héritage est complet : } R_1[K_1] = R_2[K_1] \cup R_3[K_1]
 \end{aligned}$$

Deuxième Méthode

Pour un héritage **exclusif et complet** (ajouter une classe « autres » si besoin).
 ATTENTION : à éviter si C_1 est impliquée dans d'autres associations.

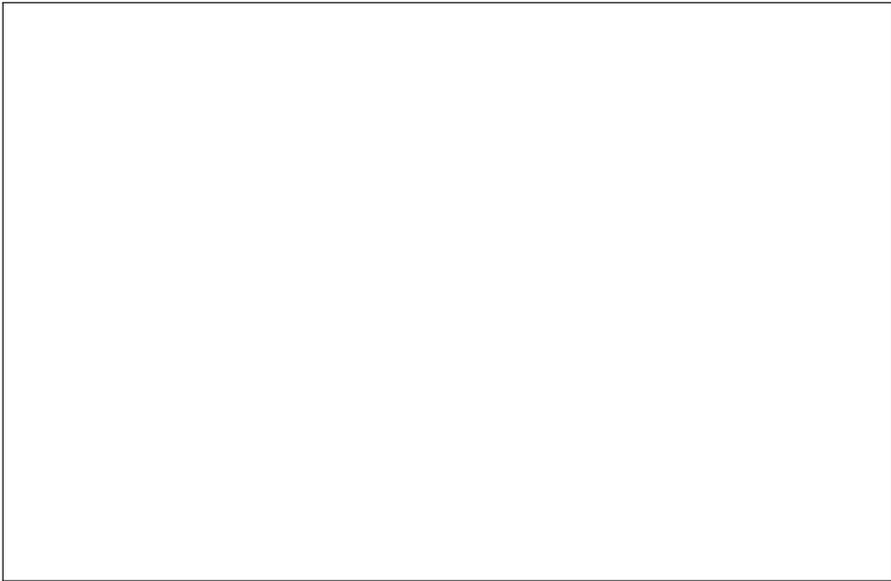
- **Pas de relation** pour C_1 .
- Une relation R_i pour toute classe C_i avec $2 \leq i \leq k$.
 - Les attributs de R_i sont les attributs de C_1 + ceux de C_i .
 - La clé primaire de R_i est K_1 , la clé de C_1 .
- Pour tout i, j avec $2 \leq i < j \leq k$ on a une contrainte $R_i[K_1] \cap R_j[K_1] = \emptyset$.

Exemple pour $k = 3$.

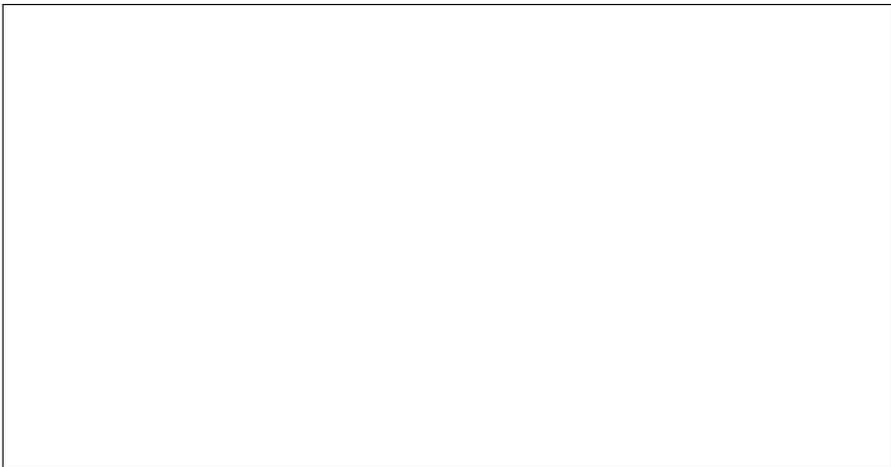


$$\begin{aligned}
 &R_2(K_1, A_1, A_2, \dots), R_3(K_1, A_1, A_3, \dots), \\
 &R_2[K_1] \cap R_3[K_1] = \emptyset
 \end{aligned}$$

Exemple



Exemple



Troisième Méthode

Pour un héritage avec **une seule sous-classe**.

- Une seule relation R_1 pour C_1 et C_2 .
- Les attributs de R_1 sont ceux de C_1 + ceux de C_2 + un attribut spécial « is- C_2 » de type Nombre.
- La valeur de « is- C_2 » est 1 si l'objet appartient à C_2 et 0 sinon.
- La clé primaire de R_1 est K_1 , la clé de C_1 .
- Pour chaque ligne de R_1 , si « is- C_2 » vaut 0 alors les attributs de C_2 ont pour valeur NULL.

Exemple pour $k = 3$.



⇒

$R_1(K_1, A_1, \text{is-}C_2, A_2, \dots)$,
 (is- $C_2 = 0$) ⇒ (A_2 est NULL)