Fondements des Bases de données : *Normalisation*

M1 - Informatique

Lhouari Nourine, Karima Ennaoui et *Simon Vilmin*.

Institut d'informatique, ISIMA

2020-2021

Les décompositions

- Une couverture Σ sur R donne un cadre pour différencier un bon d'un mauvais schéma de relation.
- Un schéma (en particulier R) et une relation peuvent contenir :
  - des *anomalies*, (définies par les DF)
  - de la *redondance*.
- Pour ôter ces problèmes intra-relation, on *décompose* R pour atteindre des schémas dits en *forme normale*, c'est la *normalisation*.
- Cependant, ces schémas doivent dans la mesure du possible *conserver* les données initiales des relations et les contraintes.
- Deux grandes formes normales : la *forme normale de Boyce-Codd* et la *3ème forme normale*.
- *Disclaimer :* d'autres formes normales utilisant plus de contraintes (multivaluées, dépendances d'inclusions) existent, mais on ne les voit pas ici.

Retour des nuages

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	cumulonimbus	gris
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir

- $R = \{Forme, Structure, Nom, Couleur\}$,
- $\Sigma = \{Forme, Structure \to Nom; Nom \to Forme; Couleur \to Structure\}$,
- Les clés sont $\{Couleur, Forme\}$ et $\{Couleur, Nom\}$.

Est-ce un bon schéma ?

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	cumulonimbus	gris
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir

Est-ce un bon schéma ?

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	cumulonimbus	gris
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir
galet	gouttelettes	cumulus	gris

✗ Le nouveau tuple *est bien identifié* par les clés, pourtant il ne *respecte pas* toutes les contraintes.

Est-ce un bon schéma ?

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	cumulonimbus	gris
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir
galet	gouttelettes	cumulus	gris

✗ Le nouveau tuple *est bien identifié* par les clés, pourtant il ne *respecte pas* toutes les contraintes.

Est-ce un bon schéma ?

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	cumulonimbus	gris
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir

- ✗ Le nouveau tuple *est bien identifié* par les clés, pourtant il ne *respecte pas* toutes les contraintes.
- ✗ Par erreur, une modification *cassant les contraintes* est apportée au troisième tuple. Celui-ci *respectant les clés*, l'erreur passe inaperçu.

Est-ce un bon schéma ?

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	altostratus	translucide
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir

✗ Le nouveau tuple *est bien identifié* par les clés, pourtant il ne *respecte pas* toutes les contraintes.

✗ Par erreur, une modification *cassant les contraintes* est apportée au troisième tuple. Celui-ci *respectant les clés*, l'erreur passe inaperçu.

Est-ce un bon schéma ?

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	altostratus	translucide
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir

- ✗ Le nouveau tuple *est bien identifié* par les clés, pourtant il ne *respecte pas* toutes les contraintes.
- ✗ Par erreur, une modification *cassant les contraintes* est apportée au troisième tuple. Celui-ci *respectant les clés*, l'erreur passe inaperçu.

Est-ce un bon schéma ?

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	cumulonimbus	gris
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir

- ✗ Le nouveau tuple *est bien identifié* par les clés, pourtant il ne *respecte pas* toutes les contraintes.
- ✗ Par erreur, une modification *cassant les contraintes* est apportée au troisième tuple. Celui-ci *respectant les clés*, l'erreur passe inaperçu.
- ✗ Le *fait* qu'un cumulonimbus ait une forme de mouton *apparaît plusieurs fois*. De même pour le fait que la couleur grise provienne de cristaux et de gouttelettes.

Est-ce un bon schéma ?

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	cumulonimbus	gris
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir

- ✗ Le nouveau tuple *est bien identifié* par les clés, pourtant il ne *respecte pas* toutes les contraintes.
- ✗ Par erreur, une modification *cassant les contraintes* est apportée au troisième tuple. Celui-ci *respectant les clés*, l'erreur passe inaperçu.
- ✗ Le *fait* qu'un cumulonimbus ait une forme de mouton *apparaît plusieurs fois*. De même pour le fait que la couleur grise provienne de cristaux et de gouttelettes.

Première alternative

Forme	Nom
couche	cirrostratus
couche	altostratus
mouton	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

Première alternative

Forme	Nom
couche	cirrostratus
couche	altostratus
mouton	cumulonimbus
galet	cumulus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir
gouttelettes	gris

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir
cumulus	gris

✓ Un tuple ne peut plus être inséré dans une des relations *sans respecter toutes les contraintes*. (relativement aux attributs de chaque relation)

Première alternative

Forme	Nom
couche	cirrostratus
couche	altostratus
mouton	cumulonimbus
galet	cumulus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir
cumulus	gris

✓ Un tuple ne peut plus être inséré dans une des relations *sans respecter toutes les contraintes*. (relativement aux attributs de chaque relation)

Première alternative

Forme	Nom
couche	cirrostratus
couche	altostratus
mouton	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

- ✓ Un tuple ne peut plus être inséré dans une des relations *sans respecter toutes les contraintes*. (relativement aux attributs de chaque relation)
- ✓ Pareil pour les mises à jour.

Première alternative

Forme	Nom
couche	cirrostratus
couche	altostratus
mouton	altostratus

Structure	Couleur
cristaux	translucide
c + g	translucide
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
altostratus	translucide
cumulonimbus	blanc
cumulonimbus	noir

- ✓ Un tuple ne peut plus être inséré dans une des relations *sans respecter toutes les contraintes*. (relativement aux attributs de chaque relation)
- ✓ Pareil pour les mises à jour.

Première alternative

Forme	Nom
couche	cirrostratus
couche	altostratus
mouton	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
altostratus	translucide
cumulonimbus	blanc
cumulonimbus	noir

- ✓ Un tuple ne peut plus être inséré dans une des relations *sans respecter toutes les contraintes*. (relativement aux attributs de chaque relation)
- ✓ Pareil pour les mises à jour.

Première alternative

Forme	Nom
couche	cirrostratus
couche	altostratus
mouton	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

- ✓ Un tuple ne peut plus être inséré dans une des relations *sans respecter toutes les contraintes*. (relativement aux attributs de chaque relation)
- ✓ Pareil pour les mises à jour.
- ✓ La redondance a disparu.

Première alternative

Forme	Nom
couche	cirrostratus
couche	altostratus
mouton	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

- ✓ Un tuple ne peut plus être inséré dans une des relations *sans respecter toutes les contraintes*. (relativement aux attributs de chaque relation)
- ✓ Pareil pour les mises à jour.
- ✓ La redondance a disparu.

Première alternative

Forme	Nom
couche	cirrostratus
couche	altostratus
mouton	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

- ✓ Un tuple ne peut plus être inséré dans une des relations *sans respecter toutes les contraintes*. (relativement aux attributs de chaque relation)
- ✓ Pareil pour les mises à jour.
- ✓ La redondance a disparu.
- ✗ On a perdu la DF *Forme, Structure* $\rightarrow$ *Nom*.

Seconde alternative

Forme	Structure	Nom
couche	cristaux	cirrostratus
couche	c + g	altostratus
mouton	c + g	cumulonimbus
mouton	gouttelettes	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

Seconde alternative

Forme	Structure	Nom
couche	cristaux	cirrostratus
couche	c + g	altostratus
mouton	c + g	cumulonimbus
mouton	gouttelettes	cumulonimbus
galet	gouttelettes	cumulus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir
gouttelettes	gris

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir
cumulus	gris

✓ A nouveau, notre problème d'insertion disparaît.

Seconde alternative

Forme	Structure	Nom
couche	cristaux	cirrostratus
couche	c + g	altostratus
mouton	c + g	cumulonimbus
mouton	gouttelettes	cumulonimbus
galet	gouttelettes	cumulus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir
cumulus	gris

✓ A nouveau, notre problème d'insertion disparaît.

Seconde alternative

Forme	Structure	Nom
couche	cristaux	cirrostratus
couche	c + g	altostratus
mouton	c + g	cumulonimbus
mouton	gouttelettes	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

- ✓ A nouveau, notre problème d'insertion disparaît.
- ✓ Pareil pour les mises à jour.

Seconde alternative

Forme	Structure	Nom
couche	cristaux	cirrostratus
couche	c + g	altostratus
mouton	c + g	altostratus
mouton	gouttelettes	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	translucide
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
altostratus	translucide
cumulonimbus	blanc
cumulonimbus	noir

- ✓ A nouveau, notre problème d'insertion disparaît.
- ✓ Pareil pour les mises à jour.

Seconde alternative

Forme	Structure	Nom
couche	cristaux	cirrostratus
couche	c + g	altostratus
mouton	c + g	cumulonimbus
mouton	gouttelettes	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
altostratus	translucide
cumulonimbus	blanc
cumulonimbus	noir

- ✓ A nouveau, notre problème d'insertion disparaît.
- ✓ Pareil pour les mises à jour.

Seconde alternative

Forme	Structure	Nom
couche	cristaux	cirrostratus
couche	c + g	altostratus
mouton	c + g	cumulonimbus
mouton	gouttelettes	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

- ✓ A nouveau, notre problème d'insertion disparaît.
- ✓ Pareil pour les mises à jour.
- ✗ En revanche, on n'a *pas éliminé toute* la redondance.

Seconde alternative

Forme	Structure	Nom
couche	cristaux	cirrostratus
couche	c + g	altostratus
mouton	c + g	cumulonimbus
mouton	gouttelettes	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

- ✓ A nouveau, notre problème d'insertion disparaît.
- ✓ Pareil pour les mises à jour.
- ✗ En revanche, on n'a *pas éliminé toute* la redondance.

Seconde alternative

Forme	Structure	Nom
couche	cristaux	cirrostratus
couche	c + g	altostratus
mouton	c + g	cumulonimbus
mouton	gouttelettes	cumulonimbus

Structure	Couleur
cristaux	translucide
c + g	gris
gouttelettes	blanc
gouttelettes	noir

Nom	Couleur
cirrostratus	translucide
altostratus	gris
cumulonimbus	gris
cumulonimbus	blanc
cumulonimbus	noir

- ✓ A nouveau, notre problème d'insertion disparaît.
- ✓ Pareil pour les mises à jour.
- ✗ En revanche, on n'a *pas éliminé toute* la redondance.
- ✓ Mais la DF *Forme, Structure* → *Nom* est *préservée* !

Avant tout : simplifions

- Pour des besoins scénaristiques, simplifions l'exemple.

Forme	Structure	Nom	Couleur
couche	cristaux	cirrostratus	translucide
couche	c + g	altostratus	gris
mouton	c + g	cumulonimbus	gris
mouton	gouttelettes	cumulonimbus	blanc
mouton	gouttelettes	cumulonimbus	noir

Avant tout : simplifions

- Pour des besoins scénaristiques, simplifions l'exemple.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

Clés (Rappel)

- Un *petit* ensemble d'attributs K qui permet d'identifier un tuple dans une relation est une *clé*.
- En termes de DF : $K \rightarrow \mathrm{R}$
- Un attribut A qui appartient à une clé est dit *premier*.

Définition - Clé (candidate), superclé

Soit Σ un ensemble de DF sur R et $K \subseteq \mathrm{R}$. On dit que K est une *clé (candidate)* et que $K \rightarrow \mathrm{R}$ est une *dépendance clé* respectivement à Σ si :

- $\Sigma \models K \rightarrow \mathrm{R}$,
- pour tout $K' \subset K$, $\Sigma \not\models K' \rightarrow \mathrm{R}$.

L'ensemble des dépendances clé de Σ est noté $\mathbb{K}_\Sigma$. Si $X \subseteq \mathrm{R}$ contient une clé, c'est une *superclé*.

Notion d'anomalie (basée DF)

- Soit R, Σ et r telle que $r \models \Sigma$.

Notion d'anomalie (basée DF)

- Soit R, Σ et r telle que $r \models \Sigma$.
- En particulier donc, $r \models \mathbb{K}_{\Sigma}$.

Notion d'anomalie (basée DF)

- Soit R, Σ et r telle que $r \models \Sigma$.
- En particulier donc, $r \models \mathbb{K}_\Sigma$.
- Un SGBD ne gère que les dépendances *clés*.

Notion d'anomalie (basée DF)

- Soit R, Σ et r telle que $r \models \Sigma$.
- En particulier donc, $r \models \mathbb{K}_\Sigma$.
- Un SGBD ne gère que les dépendances *clés*.
- Il se pourrait qu'en *modifiant* r, on *respecte toujours* $\mathbb{K}_\Sigma$, mais *plus* Σ.

Notion d'anomalie (basée DF)

- Soit R, Σ et r telle que $r \models \Sigma$.
- En particulier donc, $r \models \mathbb{K}_\Sigma$.
- Un SGBD ne gère que les dépendances *clés*.
- Il se pourrait qu'en *modifiant* r, on *respecte toujours* $\mathbb{K}_\Sigma$, mais *plus* Σ.
- Et ce, *sans* que le SGBD ne le remarque !

Notion d'anomalie (basée DF)

- Soit R, Σ et r telle que $r \models \Sigma$.
- En particulier donc, $r \models \mathbb{K}_\Sigma$.
- Un SGBD ne gère que les dépendances *clés*.
- Il se pourrait qu'en *modifiant* r, on *respecte toujours* $\mathbb{K}_\Sigma$, mais *plus* Σ.
- Et ce, *sans* que le SGBD ne le remarque !
- Alors, r contient des *anomalies*, ainsi que le schéma R.

Compatibilité d'un tuple

- Soit r une relation sur R et des DF Σ.
- Note : ici on a pas forcément $r \models \Sigma$.
- Un *tuple compatible* avec r est un tuple que l'on peut ajouter à r *sans fausser* les clés $\mathbb{K}_\Sigma$.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

$$\Sigma = \{C \rightarrow S, N \rightarrow F, FS \rightarrow N\}$$

$$\mathbb{K}_\Sigma = \{CF \rightarrow NS, CN \rightarrow FS\}$$

Compatibilité d'un tuple

- Soit r une relation sur R et des DF Σ.
- Note : ici on a pas forcément $r \models \Sigma$.
- Un *tuple compatible* avec r est un tuple que l'on peut ajouter à r *sans fausser* les clés $\mathbb{K}_\Sigma$.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3
1	2	4	3

$$\Sigma = \{C \rightarrow S, N \rightarrow F, FS \rightarrow N\}$$
$$\mathbb{K}_\Sigma = \{CF \rightarrow NS, CN \rightarrow FS\}$$

Compatibilité d'un tuple

- Soit r une relation sur R et des DF Σ.
- Note : ici on a pas forcément $r \models \Sigma$.
- Un *tuple compatible* avec r est un tuple que l'on peut ajouter à r *sans fausser* les clés $\mathbb{K}_\Sigma$.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3
1	2	4	3

$$\Sigma = \{C \rightarrow S, N \rightarrow F, FS \rightarrow N\}$$

$$\mathbb{K}_\Sigma = \{CF \rightarrow NS, CN \rightarrow FS\}$$

Compatibilité d'un tuple

- Soit r une relation sur R et des DF Σ.
- Note : ici on a pas forcément $r \models \Sigma$.
- Un *tuple compatible* avec r est un tuple que l'on peut ajouter à r *sans fausser* les clés $\mathbb{K}_\Sigma$.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3
2	3	5	4

$$\Sigma = \{C \rightarrow S, N \rightarrow F, FS \rightarrow N\}$$

$$\mathbb{K}_\Sigma = \{CF \rightarrow NS, CN \rightarrow FS\}$$

Compatibilité d'un tuple

- Soit r une relation sur R et des DF Σ.
- Note : ici on a pas forcément $r \models \Sigma$.
- Un *tuple compatible* avec r est un tuple que l'on peut ajouter à r *sans fausser* les clés $\mathbb{K}_\Sigma$.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3
2	3	5	4

$$\Sigma = \{C \rightarrow S, N \rightarrow F, FS \rightarrow N\}$$

$$\mathbb{K}_\Sigma = \{CF \rightarrow NS, CN \rightarrow FS\}$$

Compatibilité d'un tuple

- Soit r une relation sur R et des DF Σ.
- Note : ici on a pas forcément $r \models \Sigma$.
- Un *tuple compatible* avec r est un tuple que l'on peut ajouter à r *sans fausser* les clés $\mathbb{K}_\Sigma$.

Définition - Tuple compatible

Soit r une relation sur R avec les DF Σ, et t un tuple sur R. Le tuple t est *compatible* avec r (respectivement à Σ) si $r \cup \{t\} \models \mathbb{K}_\Sigma$.

Anomalies : Insertion

- On a une relation r qui satisfait Σ.
- On peut y ajouter un tuple t *compatible* avec r.
- Mais il se pourrait que certaines DF de Σ ne soient *plus satisfaites* par $r \cup \{t\}$.
- C'est une *anomalie d'insertion* de r et du *schéma* R.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

$$\Sigma = \{C \to S, N \to F, FS \to N\}$$
$$\mathbb{K}_\Sigma = \{CF \to NS, CN \to FS\}$$

Anomalies : Insertion

- On a une relation r qui satisfait Σ.
- On peut y ajouter un tuple t *compatible* avec r.
- Mais il se pourrait que certaines DF de Σ ne soient *plus satisfaites* par $r \cup \{t\}$.
- C'est une *anomalie d'insertion* de r et du *schéma* R.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3
0	1	1	2

$$\Sigma = \{C \to S, N \to F, FS \to N\}$$
$$\mathbb{K}_\Sigma = \{CF \to NS, CN \to FS\}$$

Anomalies : Insertion

- On a une relation r qui satisfait Σ.
- On peut y ajouter un tuple t *compatible* avec r.
- Mais il se pourrait que certaines DF de Σ ne soient *plus satisfaites* par $r \cup \{t\}$.
- C'est une *anomalie d'insertion* de r et du *schéma* R.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3
0	1	1	2

$$\Sigma = \{C \to S, N \to F, FS \to N\}$$

$$\mathbb{K}_\Sigma = \{CF \to NS, CN \to FS\}$$

Anomalies : Insertion

- On a une relation r qui satisfait Σ.
- On peut y ajouter un tuple t *compatible* avec r.
- Mais il se pourrait que certaines DF de Σ ne soient *plus satisfaites* par $r \cup \{t\}$.
- C'est une *anomalie d'insertion* de r et du *schéma* R.

Définition - Anomalie d'insertion

Soit r une relation sur R et Σ avec $r \models \Sigma$. On dit que r possède une *anomalie d'insertion* s'il existe un tuple t compatible avec r tel que $r \cup \{t\} \not\models \Sigma$. Le schéma R a également une *anomalie d'insertion* s'il existe une relation r sur R avec cette anomalie.

Anomalies : Modification

- On veut mettre à jour r en changeant un tuple t.
- Si en *changeant* t on *ne satisfait plus* Σ mais que l'on reste compatible avec r, r et R possèdent une *anomalie de modification*.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

$$\Sigma = \{C \to S, N \to F, FS \to N\}$$
$$\mathbb{K}_\Sigma = \{CF \to NS, CN \to FS\}$$

Anomalies : Modification

- On veut mettre à jour r en changeant un tuple t.
- Si en *changeant* t on *ne satisfait plus* Σ mais que l'on reste compatible avec r, r et R possèdent une *anomalie de modification*.

F	S	N	C
0	1	2	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

$$\Sigma = \{C \to S, N \to F, FS \to N\}$$

$$\mathbb{K}_\Sigma = \{CF \to NS, CN \to FS\}$$

Anomalies : Modification

- On veut mettre à jour r en changeant un tuple t.
- Si en *changeant* t on *ne satisfait plus* Σ mais que l'on reste compatible avec r, r et R possèdent une *anomalie de modification*.

F	S	N	C
0	1	2	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

$\Sigma = \{C \rightarrow S, N \rightarrow F, FS \rightarrow N\}$

$\mathbb{K}_\Sigma = \{CF \rightarrow NS, CN \rightarrow FS\}$

Anomalies : Modification

- On veut mettre à jour r en changeant un tuple t.
- Si en *changeant* t on *ne satisfait plus* Σ mais que l'on reste compatible avec r, r et R possèdent une *anomalie de modification*.

Définition - Anomalie de modification

Soit r une relation sur R et Σ avec $r \models \Sigma$. On dit que r possède une *anomalie de modification* s'il existe un tuple u et un tuple $t \in r$ tel que u soit compatible avec $r \setminus \{t\}$ mais $(r \setminus \{t\}) \cup \{u\} \not\models \Sigma$. Le schéma R possède alors une *anomalie de modification*.

Redondance

- Une DF $X \rightarrow A$ représente un « *fait* ».
- r est *redondante* par rapport à ce « *fait* » s'il est représenté plusieurs fois.
- En dur, on a deux tuples *distincts* de r qui ont les mêmes valeurs sur X et A.
- **Remarque :** prendre deux tuples distincts implique que X n'est pas une clé !

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

$$\Sigma = \{C \rightarrow S, N \rightarrow F, FS \rightarrow N\}$$

$$\mathbb{K}_\Sigma = \{CF \rightarrow NS, CN \rightarrow FS\}$$

Redondance

- Une DF $X \rightarrow A$ représente un « *fait* ».
- r est *redondante* par rapport à ce « *fait* » s'il est représenté plusieurs fois.
- En dur, on a deux tuples *distincts* de r qui ont les mêmes valeurs sur X et A.
- **Remarque :** prendre deux tuples distincts implique que X n'est pas une clé !

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

$$\Sigma = \{C \rightarrow S, N \rightarrow F, FS \rightarrow N\}$$

$$\mathbb{K}_\Sigma = \{CF \rightarrow NS, CN \rightarrow FS\}$$

Redondance

- Une DF $X \rightarrow A$ représente un « *fait* ».
- r est *redondante* par rapport à ce « *fait* » s'il est représenté plusieurs fois.
- En dur, on a deux tuples *distincts* de r qui ont les mêmes valeurs sur X et A.
- **Remarque :** prendre deux tuples distincts implique que X n'est pas une clé !

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

$$\Sigma = \{C \rightarrow S, N \rightarrow F, FS \rightarrow N\}$$

$$\mathbb{K}_\Sigma = \{CF \rightarrow NS, CN \rightarrow FS\}$$

Redondance

- Une DF $X \rightarrow A$ représente un « *fait* ».
- r est *redondante* par rapport à ce « *fait* » s'il est représenté plusieurs fois.
- En dur, on a deux tuples *distincts* de r qui ont les mêmes valeurs sur X et A.
- **Remarque :** prendre deux tuples distincts implique que X n'est pas une clé !

Définition - Redondance

Soit r une relation R et Σ tel que $r \models \Sigma$. La relation r a un problème de *redondance* (par rapport à Σ) s'il existe une DF $X \rightarrow A \in \Sigma$ et deux tuples distincts t_1, t_2 de r tels que $t_1[X \cup \{A\}] = t_2[X \cup \{A\}]$. Le schéma R souffre donc également de *redondance*.

Synthèse

Théorème : Soit Σ un ensemble de DF sur un schéma de relation R. Les propriétés suivantes sont équivalentes :

- R possède une anomalie d'insertion,
- R possède une anomalie de modification,
- R a un problème de redondance.

Décomposition

- On a des anomalies et de la redondance, comment faire pour les enlever ?
- *Solution* : Décomposition ! ! !
  - On découpe R en une multitude de sous-ensembles $R_1, \ldots, R_n$, $n \in \mathbb{N}$.
  - Donc, une relation r sur R est elle même *projetée* sur chacun de ces sous-schémas.
  - On note $r[R_1]$ la projection de r sur R_1, c-à-d $r[R_1] = \pi_{R_1}(r)$.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

Décomposition

- On a des anomalies et de la redondance, comment faire pour les enlever ?
- *Solution* : Décomposition ! ! !
  - On découpe R en une multitude de sous-ensembles $R_1, \ldots, R_n$, $n \in \mathbb{N}$.
  - Donc, une relation r sur R est elle même *projetée* sur chacun de ces sous-schémas.
  - On note $r[R_1]$ la projection de r sur R_1, c-à-d $r[R_1] = \pi_{R_1}(r)$.

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

Projection sur FSN

Décomposition

- On a des anomalies et de la redondance, comment faire pour les enlever ?
- *Solution* : Décomposition ! ! !
  - On découpe R en une multitude de sous-ensembles $R_1, \ldots, R_n$, $n \in \mathbb{N}$.
  - Donc, une relation r sur R est elle même *projetée* sur chacun de ces sous-schémas.
  - On note $r[R_1]$ la projection de r sur R_1, c-à-d $r[R_1] = \pi_{R_1}(r)$.

F	*S*	*N*
0	0	0
0	1	1
1	1	2
1	2	2

Décomposition

- On a des anomalies et de la redondance, comment faire pour les enlever ?
- *Solution* : Décomposition ! ! !
  - On découpe R en une multitude de sous-ensembles $R_1, \ldots, R_n$, $n \in \mathbb{N}$.
  - Donc, une relation r sur R est elle même *projetée* sur chacun de ces sous-schémas.
  - On note $r[R_1]$ la projection de r sur R_1, c-à-d $r[R_1] = \pi_{R_1}(r)$.

Définition - Décomposition

Une *décomposition* d'un schéma de relation R est un ensemble $\mathbf{R} = \{R_1, \ldots, R_n\}$, $n \in \mathbb{N}$ telle que pour tout $1 \leq i \leq n$, $R_i \subseteq R$ et telle que $\bigcup_{1 \leq i \leq n} R_i = R$.
La décomposition d'une relation r sur R par rapport à $\mathbf{R}$ est l'ensemble de relations $\{r[R_1], \ldots, r[R_n]\}$, pour $1 \leq i \leq n$.

Décomposition sans perte de données

- Une décomposition sépare une relation en d'autres, plus petites.
- Les tuples sont donc éclatés dans ces petites relations.
- **Question :** si on recolle les morceaux (par $\bowtie$), revient-t-on au point de départ ?

F	S	N
0	0	0
0	1	1
1	1	2
1	2	2

Décomposition sans perte de données

- Une décomposition sépare une relation en d'autres, plus petites.
- Les tuples sont donc éclatés dans ces petites relations.
- **Question :** si on recolle les morceaux (par $\bowtie$), revient-t-on au point de départ ?

F	S	N
0	0	0
0	1	1
1	1	2
1	2	2

$\mathbf{R} = \{FS, FN\}$

F	S
0	0
0	1
1	1
1	2

F	N
0	0
0	1
1	2

Décomposition sans perte de données

- Une décomposition sépare une relation en d'autres, plus petites.
- Les tuples sont donc éclatés dans ces petites relations.
- **Question :** si on recolle les morceaux (par $\bowtie$), revient-t-on au point de départ ?

F	S	N
0	0	0
0	1	1
1	1	2
1	2	2

$\mathbf{R} = \{FS, FN\}$

F	S
0	0
0	1
1	1
1	2

F	N
0	0
0	1
1	2

$\bowtie$

F	S	N
0	0	0
0	1	0
0	0	1
0	1	1
1	1	2
1	2	2

Décomposition sans perte de données

- Une décomposition sépare une relation en d'autres, plus petites.
- Les tuples sont donc éclatés dans ces petites relations.
- **Question :** si on recolle les morceaux (par $\bowtie$), revient-t-on au point de départ ?

F	S	N
0	0	0
0	1	1
1	1	2
1	2	2

$\mathbf{R} = \{FS, FN\}$

F	S
0	0
0	1
1	1
1	2

F	N
0	0
0	1
1	2

$\bowtie$

F	S	N
0	0	0
0	1	0
0	0	1
0	1	1
1	1	2
1	2	2

Décomposition sans perte de données

- Une décomposition sépare une relation en d'autres, plus petites.
- Les tuples sont donc éclatés dans ces petites relations.
- **Question :** si on recolle les morceaux (par $\bowtie$), revient-t-on au point de départ ?

Pas toujours ...

Définition - Décomposition sans perte de données

Une décomposition $\mathbf{R} = \{R_1, \ldots, R_n\}$ de R est dite *sans perte de données* si quelque soit la relation r sur R,

$$r = \pi_{R_1}(r) \bowtie \ldots \bowtie \pi_{R_n}(r)$$

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte ...
- Mais on a une solution, les *tableaux* !

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte ...
- Mais on a une solution, les *tableaux*!
  - Une colonne par attribut de R et une ligne par élément de **R**.

R	F	S	N	C
FS				
CN				
CS				

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte ...
- Mais on a une solution, les *tableaux* !
  - Une colonne par attribut de R et une ligne par élément de **R**.
  - On associe une *variable distinguée* a_i à chaque attribut $A_i \in$ R. Si l'élément R_j contient l'attribut A_i, on met a_i dans la case correspondante.

R	F	S	N	C
FS	a_1			
CN				
CS				

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte ...
- Mais on a une solution, les *tableaux* !
  - Une colonne par attribut de R et une ligne par élément de **R**.
  - On associe une *variable distinguée* a_i à chaque attribut $A_i \in$ R. Si l'élément R_j contient l'attribut A_i, on met a_i dans la case correspondante.

R	F	S	N	C
FS	a_1	a_2		
CN				
CS		a_2		

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte ...
- Mais on a une solution, les *tableaux* !
  - Une colonne par attribut de R et une ligne par élément de **R**.
  - On associe une *variable distinguée* a_i à chaque attribut $A_i \in$ R. Si l'élément R_j contient l'attribut A_i, on met a_i dans la case correspondante.

R	F	S	N	C
FS	a_1	a_2		
CN			a_3	
CS		a_2		

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte ...
- Mais on a une solution, les *tableaux*!
  - Une colonne par attribut de R et une ligne par élément de **R**.
  - On associe une *variable distinguée* a_i à chaque attribut $A_i \in$ R. Si l'élément R$_j$ contient l'attribut A_i, on met a_i dans la case correspondante.

R	F	S	N	C
FS	a_1	a_2		
CN			a_3	a_4
CS		a_2		a_4

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte ...
- Mais on a une solution, les *tableaux*!
  - Une colonne par attribut de R et une ligne par élément de **R**.
  - On associe une *variable distinguée* a_i à chaque attribut $A_i \in$ R. Si l'élément R_j contient l'attribut A_i, on met a_i dans la case correspondante.
  - On remplit le reste du tableau de *variables non-distinguées* uniques b_k.

R	F	S	N	C
FS	a_1	a_2	b_1	b_2
CN			a_3	a_4
CS		a_2		a_4

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte ...
- Mais on a une solution, les *tableaux* !
  - Une colonne par attribut de R et une ligne par élément de **R**.
  - On associe une *variable distinguée* a_i à chaque attribut $A_i \in$ R. Si l'élément R_j contient l'attribut A_i, on met a_i dans la case correspondante.
  - On remplit le reste du tableau de *variables non-distinguées* uniques b_k.

R	F	S	N	C
FS	a_1	a_2	b_1	b_2
CN	b_3	b_4	a_3	a_4
CS		a_2		a_4

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte ...
- Mais on a une solution, les *tableaux* !
  - Une colonne par attribut de R et une ligne par élément de **R**.
  - On associe une *variable distinguée* a_i à chaque attribut $A_i \in$ R. Si l'élément R_j contient l'attribut A_i, on met a_i dans la case correspondante.
  - On remplit le reste du tableau de *variables non-distinguées* uniques b_k.

R	F	S	N	C
FS	a_1	a_2	b_1	b_2
CN	b_3	b_4	a_3	a_4
CS	b_5	a_2	b_6	a_4

Tableaux

- Difficile de tester *à priori* si une décomposition est sans perte . . .
- Mais on a une solution, les *tableaux* !
  - Une colonne par attribut de R et une ligne par élément de **R**.
  - On associe une *variable distinguée* a_i à chaque attribut $A_i \in$ R. Si l'élément R_j contient l'attribut A_i, on met a_i dans la case correspondante.
  - On remplit le reste du tableau de *variables non-distinguées* uniques b_k.

R	F	S	N	C
FS	a_1	a_2	b_1	b_2
CN	b_3	b_4	a_3	a_4
CS	b_5	a_2	b_6	a_4

Pourquoi faire

- Le tableau est un *« template »* pour construire un tuple de la jointure.

R	F	S	N	C
FN	a_1	b_1	a_3	b_2
CN	b_3	b_4	a_3	a_4
CS	b_5	a_2	b_6	a_4

Pourquoi faire

- Le tableau est un « *template* » pour construire un tuple de la jointure. Soit t un tuple de $r[FN] \bowtie r[CN] \bowtie r[CS]$. Disons que $t = \langle a_1, a_2, a_3, a_4 \rangle$.

R	F	S	N	C
FN	a_1	b_1	a_3	b_2
CN	b_3	b_4	a_3	a_4
CS	b_5	a_2	b_6	a_4
t	a_1	a_2	a_3	a_4

Pourquoi faire

- Le tableau est un *« template »* pour construire un tuple de la jointure. Soit t un tuple de $r[FN] \bowtie r[CN] \bowtie r[CS]$. Disons que $t = \langle a_1, a_2, a_3, a_4 \rangle$.
- t est un assemblage de trois tuples $u[FN]$, $v[CN]$, $w[CS]$ qui sont des projections de tuples u, v, w de r sur la décomposition :
  - $u = \langle a_1, b_1, a_3, b_2 \rangle$ puisque $t[FN] = u[FN]$,

R	F	S	N	C	
FN	a_1	b_1	a_3	b_2	u
CN	b_3	b_4	a_3	a_4	
CS	b_5	a_2	b_6	a_4	
t	a_1	a_2	a_3	a_4	

Pourquoi faire

- Le tableau est un « *template* » pour construire un tuple de la jointure. Soit t un tuple de $r[FN] \bowtie r[CN] \bowtie r[CS]$. Disons que $t = \langle a_1, a_2, a_3, a_4 \rangle$.
- t est un assemblage de trois tuples $u[FN]$, $v[CN]$, $w[CS]$ qui sont des projections de tuples u, v, w de r sur la décomposition :
  - $u = \langle a_1, b_1, a_3, b_2 \rangle$ puisque $t[FN] = u[FN]$,
  - $v = \langle b_3, b_4, a_3, a_4 \rangle$ puisque $t[CN] = v[CN]$ (notons que $v[N] = u[N]$!)

R	F	S	N	C	
FN	a_1	b_1	a_3	b_2	u
CN	b_3	b_4	a_3	a_4	v
CS	b_5	a_2	b_6	a_4	
t	a_1	a_2	a_3	a_4	

Pourquoi faire

- Le tableau est un « *template* » pour construire un tuple de la jointure. Soit t un tuple de $r[FN] \bowtie r[CN] \bowtie r[CS]$. Disons que $t = \langle a_1, a_2, a_3, a_4 \rangle$.
- t est un assemblage de trois tuples $u[FN]$, $v[CN]$, $w[CS]$ qui sont des projections de tuples u, v, w de r sur la décomposition :
  - $u = \langle a_1, b_1, a_3, b_2 \rangle$ puisque $t[FN] = u[FN]$,
  - $v = \langle b_3, b_4, a_3, a_4 \rangle$ puisque $t[CN] = v[CN]$ (notons que $v[N] = u[N]$!)
  - $w = \langle b_5, a_2, b_6, a_4 \rangle$ puisque $t[CS] = w[CS]$.

R	F	S	N	C	
FN	a_1	b_1	a_3	b_2	u
CN	b_3	b_4	a_3	a_4	v
CS	b_5	a_2	b_6	a_4	w
t	a_1	a_2	a_3	a_4	

Pourquoi faire

- Le tableau est un « *template* » pour construire un tuple de la jointure. Soit t un tuple de $r[FN] \bowtie r[CN] \bowtie r[CS]$. Disons que $t = \langle a_1, a_2, a_3, a_4 \rangle$.
- t est un assemblage de trois tuples $u[FN]$, $v[CN]$, $w[CS]$ qui sont des projections de tuples u, v, w de r sur la décomposition :
  - $u = \langle a_1, b_1, a_3, b_2 \rangle$ puisque $t[FN] = u[FN]$,
  - $v = \langle b_3, b_4, a_3, a_4 \rangle$ puisque $t[CN] = v[CN]$ (notons que $v[N] = u[N]$!)
  - $w = \langle b_5, a_2, b_6, a_4 \rangle$ puisque $t[CS] = w[CS]$.
- Maintenant, u, v, w sont des tuples de r. Donc ils respectent Σ !

R	F	S	N	C	
FN	a_1	b_1	a_3	b_2	u
CN	b_3	b_4	a_3	a_4	v
CS	b_5	a_2	b_6	a_4	w
t	a_1	a_2	a_3	a_4	

Pourquoi faire

- Le tableau est un « *template* » pour construire un tuple de la jointure. Soit t un tuple de $r[FN] \bowtie r[CN] \bowtie r[CS]$. Disons que $t = \langle a_1, a_2, a_3, a_4 \rangle$.
- t est un assemblage de trois tuples $u[FN]$, $v[CN]$, $w[CS]$ qui sont des projections de tuples u, v, w de r sur la décomposition :
  - $u = \langle a_1, b_1, a_3, b_2 \rangle$ puisque $t[FN] = u[FN]$,
  - $v = \langle b_3, b_4, a_3, a_4 \rangle$ puisque $t[CN] = v[CN]$ (notons que $v[N] = u[N]$!)
  - $w = \langle b_5, a_2, b_6, a_4 \rangle$ puisque $t[CS] = w[CS]$.
- Maintenant, u, v, w sont des tuples de r. Donc ils respectent Σ !
  - $u[N] = v[N]$ et $N \rightarrow F$, donc $v[F] = t[F] = a_1$

R	F	S	N	C	
FN	a_1	b_1	a_3	b_2	u
CN	b_3	b_4	a_3	a_4	v
CS	b_5	a_2	b_6	a_4	w
t	a_1	a_2	a_3	a_4	

Pourquoi faire

- Le tableau est un « *template* » pour construire un tuple de la jointure. Soit t un tuple de $r[FN] \bowtie r[CN] \bowtie r[CS]$. Disons que $t = \langle a_1, a_2, a_3, a_4 \rangle$.
- t est un assemblage de trois tuples $u[FN]$, $v[CN]$, $w[CS]$ qui sont des projections de tuples u, v, w de r sur la décomposition :
  - $u = \langle a_1, b_1, a_3, b_2 \rangle$ puisque $t[FN] = u[FN]$,
  - $v = \langle b_3, b_4, a_3, a_4 \rangle$ puisque $t[CN] = v[CN]$ (notons que $v[N] = u[N]$!)
  - $w = \langle b_5, a_2, b_6, a_4 \rangle$ puisque $t[CS] = w[CS]$.
- Maintenant, u, v, w sont des tuples de r. Donc ils respectent Σ !
  - $u[N] = v[N]$ et $N \rightarrow F$, donc $v[F] = t[F] = a_1$

R	F	S	N	C	
FN	a_1	b_1	a_3	b_2	u
CN	a_1	b_4	a_3	a_4	v
CS	b_5	a_2	b_6	a_4	w
t	a_1	a_2	a_3	a_4	

Pourquoi faire

- Le tableau est un « *template* » pour construire un tuple de la jointure. Soit t un tuple de $r[FN] \bowtie r[CN] \bowtie r[CS]$. Disons que $t = \langle a_1, a_2, a_3, a_4 \rangle$.
- t est un assemblage de trois tuples $u[FN]$, $v[CN]$, $w[CS]$ qui sont des projections de tuples u, v, w de r sur la décomposition :
  - $u = \langle a_1, b_1, a_3, b_2 \rangle$ puisque $t[FN] = u[FN]$,
  - $v = \langle b_3, b_4, a_3, a_4 \rangle$ puisque $t[CN] = v[CN]$ (notons que $v[N] = u[N]$!)
  - $w = \langle b_5, a_2, b_6, a_4 \rangle$ puisque $t[CS] = w[CS]$.
- Maintenant, u, v, w sont des tuples de r. Donc ils respectent Σ !
  - $u[N] = v[N]$ et $N \to F$, donc $v[F] = t[F] = a_1$
  - $w[C] = v[C]$ et $C \to S$, donc $v[S] = t[s] = a_2$

R	F	S	N	C	
FN	a_1	b_1	a_3	b_2	u
CN	a_1	b_4	a_3	a_4	v
CS	b_5	a_2	b_6	a_4	w
t	a_1	a_2	a_3	a_4	

Pourquoi faire

- Le tableau est un « *template* » pour construire un tuple de la jointure. Soit t un tuple de $r[FN] \bowtie r[CN] \bowtie r[CS]$. Disons que $t = \langle a_1, a_2, a_3, a_4 \rangle$.
- t est un assemblage de trois tuples $u[FN]$, $v[CN]$, $w[CS]$ qui sont des projections de tuples u, v, w de r sur la décomposition :
  - $u = \langle a_1, b_1, a_3, b_2 \rangle$ puisque $t[FN] = u[FN]$,
  - $v = \langle b_3, b_4, a_3, a_4 \rangle$ puisque $t[CN] = v[CN]$ (notons que $v[N] = u[N]$!)
  - $w = \langle b_5, a_2, b_6, a_4 \rangle$ puisque $t[CS] = w[CS]$.
- Maintenant, u, v, w sont des tuples de r. Donc ils respectent Σ !
  - $u[N] = v[N]$ et $N \rightarrow F$, donc $v[F] = t[F] = a_1$
  - $w[C] = v[C]$ et $C \rightarrow S$, donc $v[S] = t[s] = a_2$

R	F	S	N	C	
FN	a_1	b_1	a_3	b_2	u
CN	a_1	a_2	a_3	a_4	v
CS	b_5	a_2	b_6	a_4	w
t	a_1	a_2	a_3	a_4	

Et donc ?

- Vu $t = v \in r$, on a *déduit* qu'un tuple t de la jointure est déjà dans r, donc la décomposition est *sans perte* !
- Implicitement, on a appliqué Σ sur le tableau en utilisant l'*algorithme CHASE* (avec $a_i \leq b_j$ et $b_j \leq b_k$ si $j \leq k$) !

R	F	S	N	C	
FN	a_1	b_1	a_3	b_2	u
CN	a_1	a_2	a_3	a_4	v
CS	b_5	a_2	b_6	a_4	w

t	a_1	a_2	a_3	a_4

Théorème : Soit Σ une couverture sur R et **R** une décomposition de R. Alors, **R** est une décomposition sans perte de données *si et seulement si* le tableau résultant de $CHASE(\Sigma, T)$ contient la ligne $\langle a_1, \ldots, a_n \rangle$ où T est le tableau associé à **R**.

Projection de DF

- Quand des DF Σ sont associées à R, elles se retrouvent aussi *projetée* par la décomposition **R**.
- La projection des DF sur $R' \subseteq R$ représente toute les DF de Σ relative aux attributs de R'.

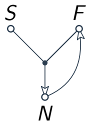
Définition - Projection d'un ensemble de DF

Soit Σ une couverture sur R et $R' \subseteq R$. La *projection* de Σ sur R', notée $\Sigma[R']$ est définie par :

$$\Sigma[R'] = \{X \to Y \mid X \to Y \in \Sigma \text{ et } X \cup Y \subseteq R'\}$$

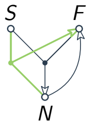
Décomposition sans perte de DF

- Quand on décompose R via **R**, on éclate l'information représentée par Σ. (ou par n'importe quelle $\Sigma' \equiv \Sigma$)
- **Rappel :** $\Sigma^+ = \{X \to Y \mid \Sigma \models X \to Y\}$.
- **Question :** si on recolle les morceaux (par $\cup$ des DF), retrouve-t-on toute l'information de Σ (donc Σ^+) ?



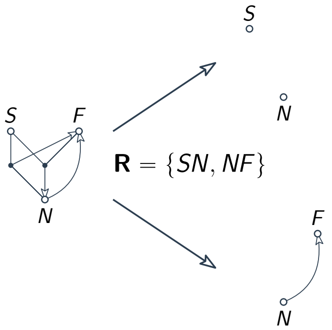
Décomposition sans perte de DF

- Quand on décompose R via **R**, on éclate l'information représentée par Σ. (ou par n'importe quelle $\Sigma' \equiv \Sigma$)
- **Rappel :** $\Sigma^+ = \{X \to Y \mid \Sigma \models X \to Y\}$.
- **Question :** si on recolle les morceaux (par $\cup$ des DF), retrouve-t-on toute l'information de Σ (donc Σ^+) ?



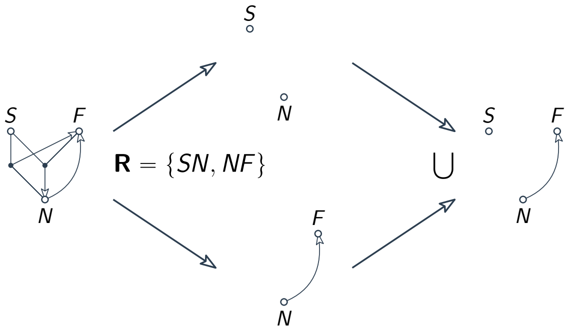
Décomposition sans perte de DF

- Quand on décompose R via **R**, on éclate l'information représentée par Σ. (ou par n'importe quelle $\Sigma' \equiv \Sigma$)
- **Rappel :** $\Sigma^+ = \{X \rightarrow Y \mid \Sigma \models X \rightarrow Y\}$.
- **Question :** si on recolle les morceaux (par $\cup$ des DF), retrouve-t-on toute l'information de Σ (donc Σ^+) ?



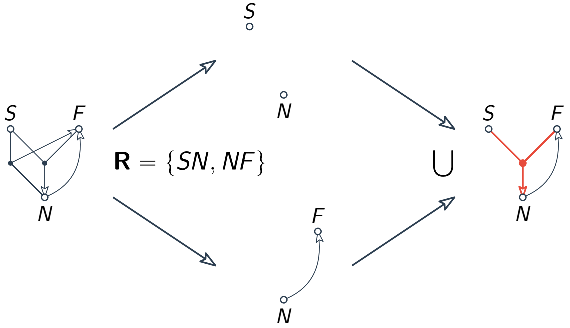
Décomposition sans perte de DF

- Quand on décompose R via **R**, on éclate l'information représentée par Σ. (ou par n'importe quelle $\Sigma' \equiv \Sigma$)
- **Rappel :** $\Sigma^+ = \{X \to Y \mid \Sigma \models X \to Y\}$.
- **Question :** si on recolle les morceaux (par $\cup$ des DF), retrouve-t-on toute l'information de Σ (donc Σ^+) ?



Décomposition sans perte de DF

- Quand on décompose R via **R**, on éclate l'information représentée par Σ. (ou par n'importe quelle $\Sigma' \equiv \Sigma$)
- **Rappel :** $\Sigma^+ = \{X \to Y \mid \Sigma \models X \to Y\}$.
- **Question :** si on recolle les morceaux (par $\cup$ des DF), retrouve-t-on toute l'information de Σ (donc Σ^+) ?



Décomposition sans perte de DF

- Quand on décompose R via **R**, on éclate l'information représentée par Σ. (ou par n'importe quelle $\Sigma' \equiv \Sigma$)
- **Rappel :** $\Sigma^+ = \{X \rightarrow Y \mid \Sigma \models X \rightarrow Y\}$.
- **Question :** si on recolle les morceaux (par $\cup$ des DF), retrouve-t-on toute l'information de Σ (donc Σ^+) ?

Pas toujours ...

Définition - Décomposition sans perte de DF

Une décomposition **R** de R est *sans perte de DF* respectivement à une couverture Σ si $\bigcup_{R_i \in \mathbf{R}} \Sigma^+[R_i] \equiv \Sigma^+$.

- Par contre pour tester, on a pas trop le choix, il faut calculer Σ^+.

Résumons

- On a un schéma de relation R et des DF sur R.
- Notre schéma R présente :
  - des anomalies,
  - des redondances.
- Pour éliminer ces problèmes, on va *décomposer* R.
- Mais ! on aimerait essayer de :
  - Préserver les DF,
  - préserver les données.

Forme normale de Boyce-Codd

- La redondance arrive quand on a une DF (non-triviale) $X \longrightarrow A$ et deux tuples *différents* d'une relation qui ont les mêmes valeurs sur X et A.

Forme normale de Boyce-Codd

- La redondance arrive quand on a une DF (non-triviale) $X \longrightarrow A$ et deux tuples *différents* d'une relation qui ont les mêmes valeurs sur X et A.
- **Remarque :** Mais si X est une (super)clé, ce problème ne peut pas arriver, puisque deux tuples *égaux sur une clé* sont *égaux tout court*.

Forme normale de Boyce-Codd

- La redondance arrive quand on a une DF (non-triviale) $X \longrightarrow A$ et deux tuples *différents* d'une relation qui ont les mêmes valeurs sur X et A.
- **Remarque :** Mais si X est une (super)clé, ce problème ne peut pas arriver, puisque deux tuples *égaux sur une clé* sont *égaux tout court*.
- **Idée :** Du coup, si *toutes* les DF non-triviales *sont des clés*, que se passe-t-il ?

Forme normale de Boyce-Codd

- La redondance arrive quand on a une DF (non-triviale) $X \rightarrow A$ et deux tuples *différents* d'une relation qui ont les mêmes valeurs sur X et A.
- **Remarque :** Mais si X est une (super)clé, ce problème ne peut pas arriver, puisque deux tuples *égaux sur une clé* sont *égaux tout court*.
- **Idée :** Du coup, si *toutes* les DF non-triviales *sont des clés*, que se passe-t-il ?

Définition - Forme Normale de Boyce-Codd (Kent)

Soit Σ une couverture sur R. On dit que le schéma R est en *forme normale de Boyce-Codd (FNBC)* (respectivement à Σ) si pour tout $X \subseteq \mathrm{R}$, $A \in \mathrm{R}$, $\Sigma \models X \rightarrow A$ implique que $\Sigma \models X \rightarrow \mathrm{R}$ ou $A \in X$. Une décomposition $\mathbf{R}$ de R est en FNBC si chacun des $\mathrm{R}' \in \mathbf{R}$ est en FNBC.

- Dans notre exemple, R n'est pas en FNBC par rapport à Σ.
- En effet, $C \rightarrow S$ est une DF non-triviale, et C n'est pas une superclé.

Propriétés

Théorème : Un schéma de relation est en FNBC *si et seulement s'* il ne souffre pas de redondance.

- La FNBC correspond à notre intuition
- faire en sorte que *« tout ne soit que clé »* *élimine la redondance* !
- En bonus : la propriété est facile à vérifier !

Propriété : R est en FNBC par rapport à Σ si et seulement si pour tout $X \rightarrow Y \in \Sigma$, X est une superclé de Σ.

Principe d'une décomposition FNBC

- On voudrait donc décomposer R en une collection de petit schémas FNBC. Pour simplifier les notations, on considère que $X \cap Y = \emptyset$ pour toute DF $X \to Y$ que l'on va traiter.

- **Approche récursive :** si $X \to Y$ ne correspond pas à une clé, on *« clétise »* X en coupant R en deux
  - $\mathrm{R}_l = X \cup Y$, du coup X sera une clé ;
  - $\mathrm{R}_r = \mathrm{R} \setminus Y$, et ici la DF $X \to Y$ n'existe plus.

  Puis on rappelle l'algo récursivement sur R_l et R_r (arbre binaire).

- ***Attention :*** pour ne pas perdre de DF en chemin, on s'appuie sur Σ^+ !

Propriété : Soit $X \to Y$ $(X \cap Y = \emptyset)$ une DF valide et $\mathrm{R}_l = X \cup Y$ et $\mathrm{R}_r = \mathrm{R} \setminus Y$. Si r est une relation sur R, alors $r = r[\mathrm{R}_l] \bowtie r[\mathrm{R}_r]$.

Décomposition FNBC

Algorithme DEC-FNBC(Σ, R)

Data: R un schéma et Σ une couverture sur R
Result: $\mathbf{R}$ une décomposition FNBC de R

Calculer Σ^+
$\mathbf{R} = \emptyset$
$\mathbf{R} = \text{DECOMPOSE}(\Sigma^+, \text{R}, \mathbf{R})$

DECOMPOSE(Σ^+, R, R)

if R *est en FNBC* **then**
  return $\mathbf{R} \cup \text{R}$
else
  Soit $X \to Y \in \Sigma^+$ telle que $X^\Sigma \neq \text{R}$
  $\text{R}_l = X \cup Y$
  $\text{R}_r = \text{R} \setminus Y$
  $\mathbf{R} = \text{DECOMPOSE}(\Sigma^+[\text{R}_l], \text{R}_l, \mathbf{R})$
  $\mathbf{R} = \text{DECOMPOSE}(\Sigma^+[\text{R}_r], \text{R}_r, \mathbf{R})$
  return $\mathbf{R}$
end

end

Trace

$$R = \{F, S, N, C\}$$

$$\Sigma = \begin{bmatrix} C \rightarrow S \\ N \rightarrow F \\ FS \rightarrow N \end{bmatrix}$$

Trace

$$R = \{F, S, N, C\}$$

$$\Sigma^{+} = \begin{bmatrix} C \rightarrow S & CN \rightarrow SF & FC \rightarrow SN \\ N \rightarrow F & SN \rightarrow F & FNC \rightarrow S \\ FS \rightarrow N & SNC \rightarrow F & FSC \rightarrow N \end{bmatrix}$$

Trace

$$R = \{F, S, N, C\}$$

$$\Sigma^+ = \begin{bmatrix} C \to S & CN \to SF & FC \to SN \\ N \to F & SN \to F & FNC \to S \\ FS \to N & SNC \to F & FSC \to N \end{bmatrix}$$

$C \to S$

$$R = \{C, S\}$$
$$\Sigma^+ = \{C \to S\}$$

$$R = \{F, N, C\}$$
$$\Sigma^+ = \{N \to F, CN \to F, FC \to N\}$$

Trace

$$R = \{F, S, N, C\}$$

$$\Sigma^+ = \begin{bmatrix} C \rightarrow S & CN \rightarrow SF & FC \rightarrow SN \\ N \rightarrow F & SN \rightarrow F & FNC \rightarrow S \\ FS \rightarrow N & SNC \rightarrow F & FSC \rightarrow N \end{bmatrix}$$

$C \rightarrow S$

$$R = \{C, S\}$$
$$\Sigma^+ = \{C \rightarrow S\}$$

$$R = \{F, N, C\}$$
$$\Sigma^+ = \{N \rightarrow F, CN \rightarrow F, FC \rightarrow N\}$$

$N \rightarrow F$

$$R = \{F, N\}$$
$$\Sigma^+ = \{N \rightarrow F\}$$

$$R = \{N, C\}$$
$$\Sigma^+ = \varnothing$$

Trace

$$R = \{F, S, N, C\}$$

$$\Sigma^+ = \begin{bmatrix} C \rightarrow S & CN \rightarrow SF & FC \rightarrow SN \\ N \rightarrow F & SN \rightarrow F & FNC \rightarrow S \\ FS \rightarrow N & SNC \rightarrow F & FSC \rightarrow N \end{bmatrix}$$

$C \rightarrow S$

$$R = \{C, S\}$$
$$\Sigma^+ = \{C \rightarrow S\}$$

$$R = \{F, N, C\}$$
$$\Sigma^+ = \{N \rightarrow F, CN \rightarrow F, FC \rightarrow N\}$$

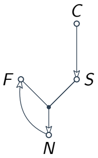
$N \rightarrow F$

$$R = \{F, N\}$$
$$\Sigma^+ = \{N \rightarrow F\}$$

$$R = \{N, C\}$$
$$\Sigma^+ = \varnothing$$

Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

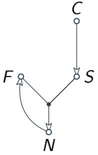


Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

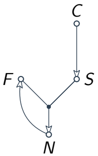
$\xrightarrow{CS}$

S	C
0	0
1	1
2	2
2	3



Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

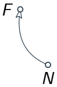


$\xrightarrow{CS}$

S	C
0	0
1	1
2	2
2	3

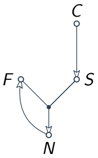
$\xrightarrow{FN}$

F	N
0	0
0	1
1	2



Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

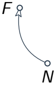


$\xrightarrow{CS}$

S	C
0	0
1	1
2	2
2	3

$\xrightarrow{FN}$

F	N
0	0
0	1
1	2

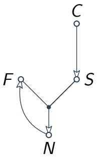


$\xrightarrow{NC}$

N	C
0	0
1	1
2	1
2	3
2	2

Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

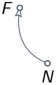


CS

S	C
0	0
1	1
2	2
2	3

FN

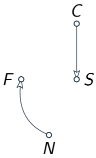
F	N
0	0
0	1
1	2



NC

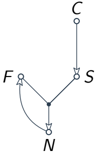
N	C
0	0
1	1
2	1
2	3
2	2

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3



Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

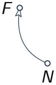


CS

S	C
0	0
1	1
2	2
2	3

FN

F	N
0	0
0	1
1	2

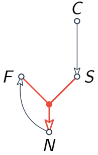


NC

N	C
0	0
1	1
2	1
2	3
2	2

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

✓



✗

Bonne et mauvaise nouvelle

Théorème : L'algorithme **DEC-FNBC**(Σ, R) calcule une décomposition sans perte de données de R en FNBC.

- **Par contre :** la décomposition obtenue *ne préserve pas* forcément les DF.
- Même pire ! Certaines paires (R, Σ) ne peuvent être décomposées en FNBC sans perdre de DF !
- **Question :** peut-on *« affaiblir »* les contraintes de la FNBC et trouver une forme normale qui préserve données et DF ?

3ème Forme Normale

- On va relâcher la contrainte *« tout le monde est une clé »*, car trop restrictive sur la préservation de DF.

3ème Forme Normale

- On va relâcher la contrainte *« tout le monde est une clé »*, car trop restrictive sur la préservation de DF.
- Pour *limiter la redondance au mieux*, on va *l'autoriser* *uniquement sur les clés*.

3ème Forme Normale

- On va relâcher la contrainte *« tout le monde est une clé »*, car trop restrictive sur la préservation de DF.
- Pour *limiter la redondance au mieux*, on va *l'autoriser* *uniquement sur les clés*.
- Intuitivement, une clé identifie un tuple, donc ses valeurs ne doivent pas trop se répéter.

3ème Forme Normale

- On va relâcher la contrainte *« tout le monde est une clé »*, car trop restrictive sur la préservation de DF.
- Pour *limiter la redondance au mieux*, on va *l'autoriser* *uniquement sur les clés*.
- Intuitivement, une clé identifie un tuple, donc ses valeurs ne doivent pas trop se répéter.

Définition - 3ème forme normale

Soit Σ une couverture sur R. Le schéma R est en *3ème forme normale (3FN)* (resp. à Σ) si pour tout $X \subseteq \mathrm{R}$, $A \in \mathrm{R} \setminus X$ si $\Sigma \models X \to A$ alors : X est une superclé, ou A est *premier* (il appartient à une clé). Une décomposition $\mathbf{R}$ de R est en 3FN si chacun des $\mathrm{R}' \in \mathbf{R}$ l'est.

- Dans notre exemple, R n'est pas en 3FN par rapport à Σ.
- En effet $C \to S$ est non-triviale, pourtant S n'est pas dans les clés CN et CF.

Propriétés

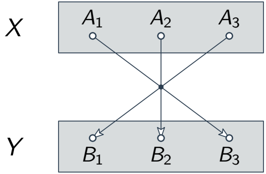
- Pourquoi 3ème ? Quid de la 1ère et la 2ème ?
- Formes intermédiaires pour la 3NF, ne nous intéresse pas ici.
- FNBC $\implies$ 3FN.

> **Propriété :** Tester qu'un schéma de relation R est en 3FN par rapport à Σ est *NP*-Complet.

- On va devoir trouver des conditions qui nous garantissent la 3FN sans tester

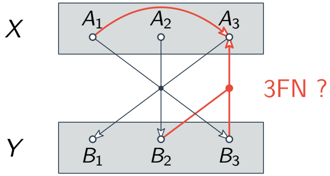
Principe d'une décomposition 3FN

- **Idée :** si on associe à chaque DF $X \rightarrow Y$ de Σ un schéma $X \cup Y$, la préservation de DF est garantie.
- Mais ! Le schéma sera-t-il en 3NF ? On pourrait avoir plusieurs soucis, supposons que l'on ait $\Sigma \models Z \rightarrow A$ avec $Z \cup A$ dans $X \cup Y$



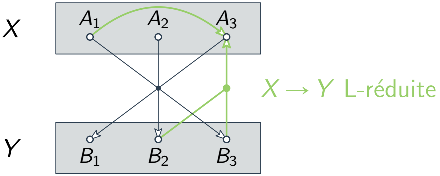
Principe d'une décomposition 3FN

- **Idée :** si on associe à chaque DF $X \rightarrow Y$ de Σ un schéma $X \cup Y$, la préservation de DF est garantie.
- Mais ! Le schéma sera-t-il en 3NF ? On pourrait avoir plusieurs soucis, supposons que l'on ait $\Sigma \models Z \rightarrow A$ avec $Z \cup A$ dans $X \cup Y$
- Si Z n'est pas une clé, A est dans X, A premier ou non ?



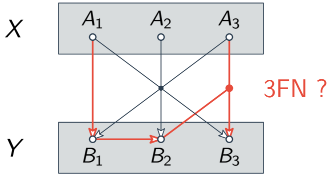
Principe d'une décomposition 3FN

- **Idée :** si on associe à chaque DF $X \rightarrow Y$ de Σ un schéma $X \cup Y$, la préservation de DF est garantie.
- Mais ! Le schéma sera-t-il en 3NF ? On pourrait avoir plusieurs soucis, supposons que l'on ait $\Sigma \models Z \rightarrow A$ avec $Z \cup A$ dans $X \cup Y$
- Si Z n'est pas une clé, A est dans X, A premier ou non ?
- Si $X \rightarrow Y$ est *L-réduite*, X devient une *clé candidate* de $X \cup Y$, répondant au problème.



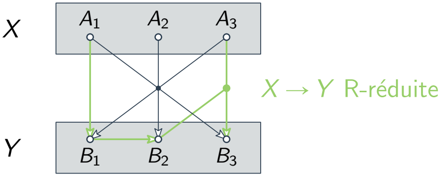
Principe d'une décomposition 3FN

- **Idée :** si on associe à chaque DF $X \rightarrow Y$ de Σ un schéma $X \cup Y$, la préservation de DF est garantie.
- Mais ! Le schéma sera-t-il en 3NF ? On pourrait avoir plusieurs soucis, supposons que l'on ait $\Sigma \models Z \rightarrow A$ avec $Z \cup A$ dans $X \cup Y$
- Si Z n'est pas une clé, A est dans Y, A premier ou non ?



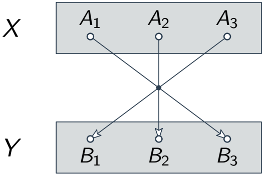
Principe d'une décomposition 3FN

- **Idée :** si on associe à chaque DF $X \to Y$ de Σ un schéma $X \cup Y$, la préservation de DF est garantie.
- Mais ! Le schéma sera-t-il en 3NF ? On pourrait avoir plusieurs soucis, supposons que l'on ait $\Sigma \models Z \to A$ avec $Z \cup A$ dans $X \cup Y$
- Si Z n'est pas une clé, A est dans Y, A premier ou non ?
- Puisque X est une clé, on peut alors *R-réduire* $X \to Y$ pour faire disparaître le problème.



Principe d'une décomposition 3FN

- **Idée :** si on associe à chaque DF $X \rightarrow Y$ de Σ un schéma $X \cup Y$, la préservation de DF est garantie.
- Mais ! Le schéma sera-t-il en 3NF ? On pourrait avoir plusieurs soucis, supposons que l'on ait $\Sigma \models Z \rightarrow A$ avec $Z \cup A$ dans $X \cup Y$
- Donc, on utilise une couverture *réduite* !
- Pour limiter le nombre de sous-relations, on *minimise* aussi Σ



Décomposition 3FN

Algorithme DEC-3FN(Σ, R)

Data: R un schéma et Σ une couverture sur R
Result: $\mathbf{R}$ une décomposition 3FN de R

Minimiser et réduire Σ
$\mathbf{R} = \emptyset$
for $X \rightarrow Y \in \Sigma$ **do**
  $\mathbf{R} = \mathbf{R} \cup \{X \cup Y\}$
end
if $\mathbf{R}$ *ne préserve pas les données* **then**
  Trouver une clé K de Σ
  $\mathbf{R} = \mathbf{R} \cup \{K\}$
end
return $\mathbf{R}$

Trace

$$\Sigma$$

$$C \rightarrow S$$

$$N \rightarrow F$$

$$SF \rightarrow N$$

Trace

Σ		**R**
$C \rightarrow S$	$\longrightarrow$	$\{C, S\}$
$N \rightarrow F$	$\longrightarrow$	$\{N, F\}$
$SF \rightarrow N$	$\longrightarrow$	$\{F, S, N\}$

Trace

Σ		**R**
$C \rightarrow S$	$\longrightarrow$	$\{C, S\}$
	redondant	
$N \rightarrow F$	$\longrightarrow$	$\{N, F\}$
$SF \rightarrow N$	$\longrightarrow$	$\{F, S, N\}$

Trace

Σ		$\mathbf{R}$
$C \rightarrow S$	$\longrightarrow$	$\{C, S\}$
$N \rightarrow F$		
$SF \rightarrow N$	$\longrightarrow$	$\{F, S, N\}$

$\longrightarrow$

	F	S	N	C
CS	b_1	a_2	b_2	a_4
FSN	a_1	a_2	a_3	b_3

Trace

Σ		$\mathbf{R}$
$C \rightarrow S$	$\longrightarrow$	$\{C, S\}$
$N \rightarrow F$		
$SF \rightarrow N$	$\longrightarrow$	$\{F, S, N\}$

$\longrightarrow$

	F	S	N	C
CS	b_1	a_2	b_2	a_4
FSN	a_1	a_2	a_3	b_3
CN	b_4	b_5	a_3	a_4

Trace

Σ		$\mathbf{R}$
$C \rightarrow S$	$\longrightarrow$	$\{C, S\}$
$N \rightarrow F$		
$SF \rightarrow N$	$\longrightarrow$	$\{F, S, N\}$

$\longrightarrow$

	F	S	N	C
CS	b_1	a_2	b_2	a_4
FSN	a_1	a_2	a_3	b_3
CN	a_1	a_2	a_3	a_4

Trace

Σ		$\mathbf{R}$
$C \rightarrow S$	$\longrightarrow$	$\{C, S\}$
$N \rightarrow F$	$\longrightarrow$	
$SF \rightarrow N$	$\longrightarrow$	$\{F, S, N\}$

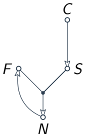
	F	S	N	C
CS	b_1	a_2	b_2	a_4
FSN	a_1	a_2	a_3	b_3
CN	a_1	a_2	a_3	a_4

$$\downarrow$$

$$\mathbf{R} = \{CS, FSN, CN\}$$

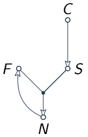
Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3



Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

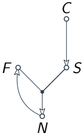


$\xrightarrow{CS}$

S	C
0	0
1	1
2	2
2	3

Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

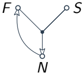


$\xrightarrow{CS}$

S	C
0	0
1	1
2	2
2	3

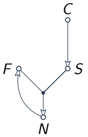
$\xrightarrow{FSN}$

F	S	N
0	0	0
0	1	1
1	1	2
1	2	2



Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

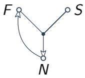


CS

S	C
0	0
1	1
2	2
2	3

FSN

F	S	N
0	0	0
0	1	1
1	1	2
1	2	2

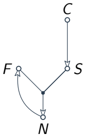


NC

N	C
0	0
1	1
2	1
2	3
2	2

Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3



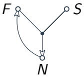
CS

FSN

NC

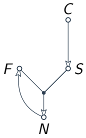
S	C
0	0
1	1
2	2
2	3

F	S	N
0	0	0
0	1	1
1	1	2
1	2	2



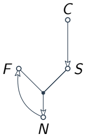
N	C
0	0
1	1
2	1
2	3
2	2

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3



Sur notre relation

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

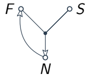


CS

S	C
0	0
1	1
2	2
2	3

FSN

F	S	N
0	0	0
0	1	1
1	1	2
1	2	2

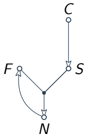


NC

N	C
0	0
1	1
2	1
2	3
2	2

F	S	N	C
0	0	0	0
0	1	1	1
1	1	2	1
1	2	2	2
1	2	2	3

✓



✓

Bonne et mauvaise nouvelle

Théorème : L'algorithme **DEC-3FN**(Σ, R) calcule une décomposition de R en 3FN qui est sans perte de données et sans perte de DF.

- Puisque 3NF n'est pas FNBC on n'*élimine pas toutes les redondances*.
- Moins complexe à calculer qu'une décomposition FNBC.

Résumé

- Au commencement, un schéma de relation et des DF.

Résumé

- Au commencement, un schéma de relation et des DF.
- avec des *anomalies* et de la *redondance* !

	Anom/Red		

Résumé

- Au commencement, un schéma de relation et des DF.
- avec des *anomalies* et de la *redondance* !
- Puis vinrent les *formes normales*, FNBC, 3FN.

	Anom/Red		
FNBC			
3FN			

Résumé

- Au commencement, un schéma de relation et des DF.
- avec des *anomalies* et de la *redondance* !
- Puis vinrent les *formes normales*, FNBC, 3FN.
- Seulement, enlever anomalies et redondance ne suffisait pas, il fallait également *préserver DF et données*.

	Anom/Red	Préserv. Donn.	Préserv. FD
FNBC			
3FN			

Résumé

- Au commencement, un schéma de relation et des DF.
- avec des *anomalies* et de la *redondance* !
- Puis vinrent les *formes normales*, FNBC, 3FN.
- Seulement, enlever anomalies et redondance ne suffisait pas, il fallait également *préserver DF et données*.
- Les deux formes normales se *partageaient* les propriétés, sans que l'une domine l'autre (même si FNBC $\geq$ 3FN dans l'absolu).

	Anom/Red	Préserv. Donn.	Préserv. FD
FNBC	✓	✓	✗
3FN	✗	✓	✓