École nationale supérieure d'informatique et de mathématiques appliquées

Dépendances Fonctionnelles et Formes Normales

Equipe pédagogique BD

2021-2022

Vision globale du cours

- Introduction SGBD et modèles de données ✓
- Bases de données relationnelles ✓
  - Modèle relationnel ✓
  - Algèbre relationnelle ✓
  - SQL ✓
- Transactions ✓
- Conception de bases de données
  - Dépendances fonctionnelles et normalisation
  - Analyse statique ✗
  - Modèle Entité-Associations ✗
  - Traduction en relationnel ✗

Une relation

Que penser de la relation suivante ?

Notes	cours	vol	nom	prénom	e-mail	note	ratt
	sport	20	Dark	Vador	vador@ensimag.fr	20	non
	sport	20	Jabba	The Hutt	jabba@tatooine.ta	3	oui
	pilotage	40	Han	Solo	solo@falcon.com	15	non
	sport	20	Han	Solo	solo@falcon.com	15	non

» *Vol* indique le volume horaire de la matière

» *Ratt* indique si l'étudiant doit rattraper la matière ou non (note <10)

Les anomalies

Notes	cours	vol	nom	prénom	e-mail	note	ratt
	sport	20	Dark	Vador	vador@ensimag.fr	20	non
	sport	20	Jabba	The Hutt	jabba@tatooine.ta	3	oui
	pilotage	40	Han	Solo	solo@falcon.com	15	non
	sport	20	Han	Solo	solo@falcon.com	15	non

- L'information est **redondante**…
- …Cela peut conduire à des anomalies :
  - de mise à jour (e.g. e-mail de Han Solo)
  - d'insertion (e.g. insertion d'une note de pilotage avec volume incohérent)
  - de suppression (e.g. suppression de la note de pilotage de Han Solo $\Rightarrow$ plus d'info sur le volume du cours)

Des redondances aux DF

Pourquoi l'information est-elle redondante ?

Parce que :

- Chaque cours a un volume unique. Si un cours apparaît plusieurs fois, son volume apparaîtra autant de fois
- Chaque étudiant a un e-mail unique. Si un étudiant apparaît plusieurs fois, son e-mail apparaîtra autant de fois.
- Etc.

Comment le sait-on ?

Parce que nous connaissons le **domaine d'application**.

La décomposition

Bon… Que faire ?

Une première idée : décomposer.

Notes1	cours	vol	nom	prénom	e-mail
	sport	20	Dark	Vador	vador@ensimag.fr
	sport	20	Jabba	The Hutt	jabba@tatooine.ta
	pilotage	40	Han	Solo	solo@falcon.com
	sport	20	Han	Solo	solo@falcon.com

Notes2	e-mail	note	ratt
	vador@ensimag.fr	20	non
	jabba@tatooine.ta	3	oui
	solo@falcon.com	15	non
	solo@falcon.com	15	non

La décomposition

Essayons de recomposer la relation initiale par jointure de ces deux relations :

Notes	cours	vol	nom	prénom	e-mail	note	Ratt
	sport	20	Dark	Vador	vador@ensimag.fr	20	Non
	sport	20	Jabba	The Hutt	jabba@tatooine.ta	3	Oui
	pilotage	40	Han	Solo	solo@falcon.com	15	Non
	pilotage	40	Han	Solo	solo@falcon.com	15	Non
	sport	20	Han	Solo	solo@falcon.com	15	Non
	sport	20	Han	Solo	solo@falcon.com	15	Non

Comparons à la relation initiale :

Notes	cours	vol	nom	prénom	e-mail	note	ratt
	sport	20	Dark	Vador	vador@ensimag.fr	20	non
	sport	20	Jabba	The Hutt	jabba@tatooine.ta	3	oui
	pilotage	40	Han	Solo	solo@falcon.com	15	non
	sport	20	Han	Solo	solo@falcon.com	15	non

Nous avons perdu une information : le lien entre les notes et les étudiants !

Conclusion : on ne peut donc pas décomposer n'importe comment…

Objectifs du cours

1. Savoir identifier les redondances potentielles

2. Caractériser la quantité d’anomalies dans une relation

3. Découper correctement une relation si besoin

Dépendance Fonctionnelle : définition

Dans l'exemple précédent, les relations entre attributs (cours/volume, prénom/nom/e-mail) sont des exemples de **dépendances fonctionnelles**.

Une relation **R** vérifie la dépendance fonctionnelle $X \rightarrow Y$ si et seulement si :

$$\forall (t_1, t_2) \in R^2, \pi_X(t_1) = \pi_X(t_2) \Rightarrow \pi_Y(t_1) = \pi_Y(t_2)$$

On dit aussi :

- *X* détermine *Y*
- *Y* dépend fonctionnellement de *X*

Pour savoir quelles DF vérifie une relation, il faut s'appuyer sur une **analyse** du domaine d'application

Exemple

Dans la relation :

Notes	cours	vol	nom	prénom	e-mail	note	ratt
	sport	20	Dark	Vador	vador@ensimag.fr	20	non
	sport	20	Jabba	The Hutt	jabba@tatooine.ta	3	oui
	pilotage	40	Han	Solo	solo@falcon.com	15	non
	sport	20	Han	Solo	solo@falcon.com	15	non

Les DF suivantes sont intuitivement vérifiées tout le temps :

- nom, prénom $\rightarrow$ e-mail
- cours, nom, prénom $\rightarrow$ note
- e-mail $\rightarrow$ prénom, nom
- cours $\rightarrow$ vol
- note $\rightarrow$ ratt

La DF prénom $\rightarrow$ nom est vérifiée dans l'extension particulière de **Notes** ci-dessus, mais **pas en général** !

Passage à la pratique

On considère une entreprise organisée de la manière suivante : elle est découpée en plusieurs départements identifiés chacun par un numéro *ND* unique.
Chaque département est caractérisé par un budget *DBUD* et par le numéro de l'employé qui assure la direction de ce département *MATD*. Chaque département emploie plusieurs employés mais on fait l'hypothèse qu'un employé ne travaille que dans un seul département. De la même façon, un département gère plusieurs projets et occupe plusieurs bureaux.
Pour chaque employé, on conserve des informations sur son numéro (unique) noté *NE*, le numéro du projet *NP* sur lequel il travaille, son numéro de bureau *NB* et son numéro de téléphone *NT*. De plus, on conserve le nom de chaque poste occupé par l'employé (*NJ* pour nom de job), et pour chacun de ces postes, la date *Date* et le salaire *SAL* correspondant.
Chaque projet décrit par un numéro unique *NP* et est caractérisé par un budget *PBud*.
Chaque bureau est caractérisé par un numéro unique *NB* et par une surface en m2 *SURF* ainsi que par tous les numéros des téléphones qu'il contient.

Conséquence logique

Reprenons l'exemple de la relation **Notes**…

Nous avons :

- e-mail $\rightarrow$ prénom, nom
- nom, prénom, cours $\rightarrow$ note

Mais e-mail, cours $\rightarrow$ note est également **automatiquement** vérifiée.
Pourquoi ?

Parce que c'est une **conséquence logique** des deux premières DF.
Preuve :
$\forall (t_1, t_2) \in \mathbf{Notes}^2$:

$$\pi_{\text{e-mail,cours}}(t_1) = \pi_{\text{e-mail,cours}}(t_2) \Rightarrow \pi_{\text{prénom,nom,cours}}(t_1) = \pi_{\text{prénom,nom,cours}}(t_2)$$
$$\Rightarrow \pi_{\text{note}}(t_1) = \pi_{\text{note}}(t_2).$$

Conséquence logique : définition

Formellement : la DF f est une **conséquence logique** d'un ensemble de DF F ssi pour toute relation **R**, on a :

$$\mathbf{R} \text{ vérifie } F \Rightarrow \mathbf{R} \text{ vérifie } f.$$

Deux ensembles de DF F et F' sont **logiquement équivalents** ssi toutes les DF de F sont des conséquences logiques de F' et vice-versa.

La **fermeture transitive** d'un ensemble F de DF est l'ensemble :

$$F^+ = \{ f \mid F \text{ implique } f \}$$

Ce sont toutes les DF que l'on peut déduire de F.

Système déductif d'Armstrong

Comment calculer les conséquences logiques ?

En utilisant le système formel d'**Armstrong**.
Règles de base :

- **Axiome de réflexivité :** si $X \subseteq Y$ alors $Y \rightarrow X$ est vérifiée
- **Règle d'augmentation :** $X \rightarrow Y$ implique $X,Z \rightarrow Y,Z$
- **Règle de transitivité :** si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$

Règles déduites :

- **Décomposition :** si $Z \subseteq Y$ alors $X \rightarrow Y$ implique $X \rightarrow Z$
- **Union :** si $X \rightarrow Y$ et $X \rightarrow Z$ alors $X \rightarrow Y,Z$
- **Pseudo-transitivité :** si $X \rightarrow Y$ et $W,Y \rightarrow Z$ alors $W,X \rightarrow Z$

Exercice : prouver les règles déduites à partir des règles de base.

Des DF sans redondance

Vous l'avez compris, il existe plein d'ensembles de DF logiquement équivalents.

S'il est logiquement équivalent de raisonner avec tous ces ensembles de DF (car ils contiennent la même information sémantique), certains sont plus intéressants que d'autres…

- La fermeture transitive F^+ (déjà vue)
- La **couverture irredondante** (ou **minimale**, car elle ne peut être réduite) **irr(*F*)**

DF élémentaire

Dans notre exemple, que penser de la DF *cours,e-mail* $\rightarrow$ *vol* ?

- Elle est parfaitement vérifiée.
- La partie gauche contient trop d'informations (*e-mail* n'est pas nécessaire).

Une telle DF n'est pas **élémentaire**.

Une DF $X \rightarrow Y$ est dite **élémentaire** dans F ssi :

1. Y ne contient qu'un attribut (elle est **canonique**)
2. $Y \not\subseteq X$ (elle est **non triviale**)
3. Il n'existe aucune DF $(X' \rightarrow Y) \in F$ telle que $X' \subset X$ (elle est **minimale**)

Couverture irredondante

Une **couverture irredondante** (ou **couverture minimale**) d'un ensemble F de DF est un ensemble noté irr(F) tel que :

- $\forall f \in$ irr(F), f est **élémentaire**
- $(\text{irr}(F))^+ = F^+$ (**équivalence logique**)
- $\forall f \in$ irr(F), $(\text{irr}(F) \setminus \{f\})^+ \neq F^+$ (f est **indispensable**)

Il existe toujours une couverture irredondante, mais elle n'est en général pas unique.

Algorithme de calcul d'une couverture irredondante

Remarque liminaire : Savoir appliquer cet algorithme est indispensable pour la suite

Pour calculer une couverture irredondante de F, on procède en trois étapes :

1. Mise sous forme **canonique** :
   $X \rightarrow y_1, y_2, \ldots, y_n$ est transformé en $X \rightarrow y_1$; $X \rightarrow y_2$; … ; $X \rightarrow y_n$
2. Mise sous forme **élémentaire** :
   - Pour chaque DF $x_1, x_2, \ldots, x_n \rightarrow y$
   - Pour chaque attribut x_i
   - Si $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_n \rightarrow y$ (la même DF sans x_i) est encore vérifiée
     alors enlever x_i de la DF (car **x_i n'est pas nécessaire**)
3. Elimination des DF **redondances** :
   - Pour chaque DF f
   - Si on peut retrouver f à partir de $F \setminus \{f\}$, alors f est redondante $\Rightarrow$ on l'enlève.

Couverture minimale : exemple

F = {(a → b,c) ; (a,b → d) ; (d → c)}.

Etape 1 : mise sous forme canonique :

- a → b
- a → c
- a,b → d
- d → c

Etape 2 : mise sous forme élémentaire :

- La seule DF potentiellement élémentaire est (a,b → d)
- Or :
  - (a → b), et par augmentation on a (a → a,b)
  - (a → a,b) et (a,b → d), et donc par transitivité on a (a → d)
- (a,b → d) n'est donc pas élémentaire (b est en trop). On la remplace donc par (a → d)

Etape 3 : élimination des redondances :

- Il y a deux chemins pour déterminer c : (a → c) et (d → c)
  or (a → c) peut être obtenue par transitivité avec (a → d) et (d → c), elle est donc redondante !

On a donc : **irr(F) = {(a → b) ; (a → d) ; (d →c)}**

Clefs d'une relation

Avant d'aborder les formes normales, nous avons besoin de deux définitions complémentaires :

Une **clef** d'une relation *R* associée à un ensemble *F* de DF est un ensemble d'attributs *X* tel que :

- $X \rightarrow y$ est une conséquence logique de *F* pour tout attribut *y* de *R*
  - $\nexists X' \subset X$ tel que $X' \rightarrow y$ est une conséquence logique de *F* pour tout attribut *y* de R
  (***X* est minimal au sens de l'inclusion**)

Attribut clef = tout attribut apparaissant dans au moins une clef.

Attribut non clef = tout attribut n'apparaissant dans aucune clef.

Exemple :

R(nom,prénom,e-mail,cours,note,vol,ratt)

F = {(nom,prénom $\rightarrow$ e-mail) ; (cours,nom,prénom $\rightarrow$ note) ; (e-mail $\rightarrow$ nom,prénom) ; (cours $\rightarrow$ vol) ; (note $\rightarrow$ ratt)}

- **Clefs :** {{cours,nom,prénom} ; {cours,e-mail}}
- **Attributs clef :** {cours, prénom, nom, e-mail}
- **Attributs non clef :** {vol, note, ratt}

Ce que nous avons vu

Jusqu'ici, nous avons vu :

- qu'une mauvaise conception des schémas pouvait entraîner des anomalies
- que ces anomalies étaient liées aux DF
- que pour caractériser correctement ces anomalies, il fallait caractériser l'ensemble des DF
- que plusieurs ensembles de DF pouvaient être équivalents, mais que pouvions travailler avec la couverture irredondante

Maintenant, nous allons voir :

- Comment caractériser formellement le degré de normalité d'une relation.
- Comment la décomposer si besoin.

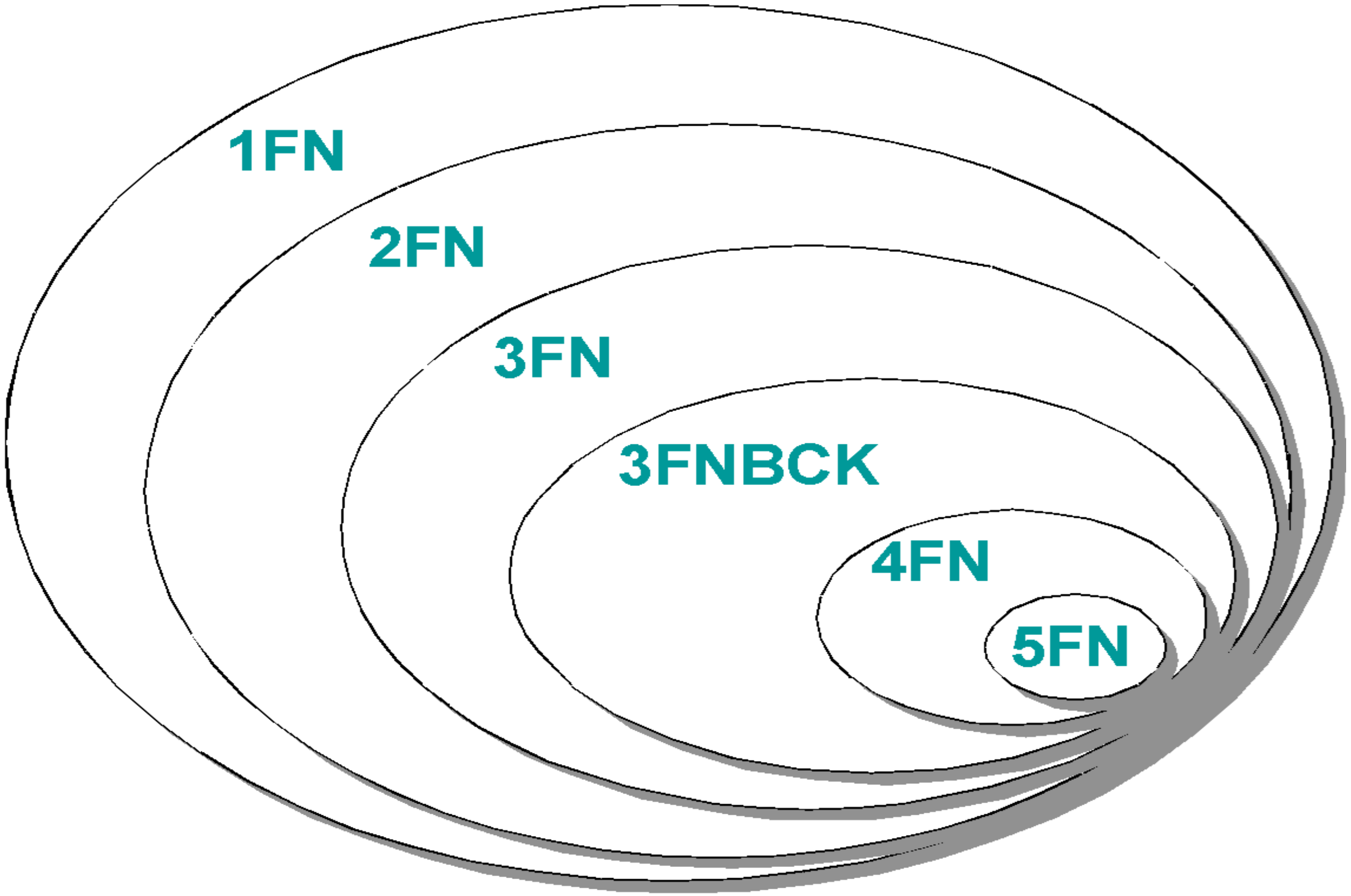
Dépendance pleine et partielle

Y est **pleinement dépendant** de X dans F ssi $X \rightarrow Y$ est une DF élémentaire de F
(sinon Y n'est que **partiellement dépendant** de X)

Exemple :
vol est pleinement dépendant de *cours* mais n'est que partiellement dépendant de $\{cours, nom\}$.

Les formes normales

Les formes normales : Une échelle ascendante de critères de normalité…



Première forme normale

Une relation *R* est en première forme normale (**1FN**) si et seulement si tout attribut ne peut contenir que des valeurs atomiques et **non Null** (Null n'est pas une valeur !)

Contre-exemple :

Elèves	prénom	nom	e-mails
	Luke	Skywalker	skywalker@ensimag.fr ; sky@gmail.com
	Dark	Vador	vador@ensimag.fr ; sky_2@gmail.com ; contact@deathstar.com

Un bon indicateur : si vous avez besoin d'un analyseur syntaxique (*parser*) pour interpréter vos valeurs d'attribut, la 1FN est violée.

Deuxième forme normale

Une relation R (et ses DF F) est en deuxième forme normale (**2FN**) si et seulement si :

1. Elle est en première forme normale (1FN);
2. Tous les attributs non clefs sont pleinement dépendants de chacune des clefs.

Exemple :

R(nom,prénom,e-mail,cours,note,vol,ratt)

F = {(nom,prénom $\rightarrow$ e-mail) ; (cours,nom,prénom $\rightarrow$ note) ; (e-mail $\rightarrow$ prénom,nom) ; (cours $\rightarrow$ vol) ; (note $\rightarrow$ ratt)}

R est-elle 2FN ?

Deuxième forme normale

Exemple de passage en 2FN :

Notes	cours	vol	nom	prénom	e-mail	note	ratt
	sport	20	Dark	Vador	vador@ensimag.fr	20	non
	sport	20	Jabba	The Hutt	jabba@tatooine.ta	3	oui
	pilotage	40	Han	Solo	solo@falcon.com	15	non
	sport	20	Han	Solo	solo@falcon.com	15	non

Volumes	cours	vol
	sport	20
	pilotage	40

Notes	cours	nom	prénom	e-mail	note	ratt
	sport	Dark	Vador	vador@ensimag.fr	20	non
	sport	Jabba	The Hutt	jabba@tatooine.ta	3	oui
	pilotage	Han	Solo	solo@falcon.com	15	non
	sport	Han	Solo	solo@falcon.com	15	non

Troisième forme normale

Une relation R (et ses DF F) est en troisième forme normale (**3FN**) si et seulement si :

1. Elle est 2FN
2. Tout attribut non clef ne dépend pas d'un ensemble d'attributs qui n'est pas une clef.

Exemple :

R(nom,prénom,e-mail,cours,note,ratt)

F = {(nom,prénom $\rightarrow$ e-mail) ; (cours,nom,prénom $\rightarrow$ note) ; (e-mail $\rightarrow$ nom,prénom) ; (note $\rightarrow$ ratt)}

R est-elle 3FN ?

Troisième forme normale

Autre définition équivalente :

Une relation R (et ses DF F) est en troisième forme normale (**3FN**) si et seulement si **tout attribut non clef est pleinement et directement dépendant de toutes les clefs**.

La **dépendance directe** s'oppose à la **dépendance transitive** définie comme suit :

Y **dépend transitivement** de X si et seulement si il existe Z tel que :

$$X \rightarrow Z \text{ et } Z \rightarrow Y \text{ avec } Z \nrightarrow X$$

Troisième forme normale

Exemple de passage en 3FN :

Notes	cours	nom	prénom	e-mail	note	ratt
	sport	Dark	Vador	vador@ensimag.fr	20	non
	sport	Jabba	The Hutt	jabba@tatooine.ta	3	oui
	pilotage	Han	Solo	solo@falcon.com	15	non
	sport	Han	Solo	solo@falcon.com	15	non

Notes	cours	nom	prénom	e-mail	note
	sport	Dark	Vador	vador@ensimag.fr	20
	sport	Jabba	The Hutt	jabba@tatooine.ta	3
	pilotage	Han	Solo	solo@falcon.com	15
	sport	Han	Solo	solo@falcon.com	15

Notes	note	ratt
	20	non
	3	oui
	15	non

Mais …

Notes	cours	nom	prénom	e-mail	note
	sport	Dark	Vador	vador@ensimag.fr	20
	~~sport~~	~~Jabba~~	~~The Hutt~~	~~jabba@tatooine.ta~~	~~3~~
	pilotage	Han	Solo	solo@falcon.com	15
	sport	Han	Solo	solo@falcon.com	15

Anomalie de suppression !

On perd l'adresse e-mail de Jabba the Hutt.

Forme normale de Boyce-Codd-Kent

Une relation R (et ses DF F) est en forme normale de Boyce-Codd-Kent (**3FNBCK**) si et seulement si :

1. Elle est 3FN
2. Pour toute DF ($X \rightarrow Y$) non triviale (i.e. $Y \nsubseteq X$), X contient une clef de R.

Exemple :

R(nom,prénom,e-mail,cours,note)

F = {(nom,prénom $\rightarrow$ e-mail) ; (cours,nom,prénom $\rightarrow$ note) ; (e-mail $\rightarrow$ nom,prénom)}

R est-elle 3FNBCK ?

Forme normale de Boyce-Codd-Kent

Exemple de passage en 3FNBCK :

Notes	cours	nom	prénom	e-mail	note
	sport	Dark	Vador	vador@ensimag.fr	20
	sport	Jabba	The Hutt	jabba@tatooine.ta	3
	pilotage	Han	Solo	solo@falcon.com	15
	sport	Han	Solo	solo@falcon.com	15

Etudiants	nom	prénom	e-mail
	Dark	Vador	vador@ensimag.fr
	Jabba	The Hutt	jabba@tatooine.ta
	Han	Solo	solo@falcon.com

Notes	cours	e-mail	note
	sport	vador@ensimag.fr	20
	sport	jabba@tatooine.ta	3
	pilotage	solo@falcon.com	15
	sport	solo@falcon.com	15

Normalisation de relations

Maintenant que nous savons caractériser le degré d'anomalies d'une relation à l'aune de ses DF, nous allons voir comment décomposer une relation qui n'est pas 3FN…

Idée :

- $R \rightsquigarrow R_1, R_2, \ldots, R_n$
- Telle que les R_i sont toutes au moins 3FN
- $R = R_1 \bowtie R_2 \bowtie \ldots \bowtie R_n$ (décomposition **sans perte d'information (SPI)**)
- Et si possible aucune DF initiale n'a disparue dans la décomposition (décomposition **sans perte de DF (SPD)**) : $F^+ = (F_1 \cup F_2 \cup \ldots \cup F_n)^+$

Deux approches possibles :

1. **Décomposer** la relation initiale *R* en s'appuyant sur les DF
2. **Synthétiser** les relations finales $R_1 \ldots R_n$ à partir des DF

Algorithme de décomposition

Le principe de l'algorithme est simple :

1. Tant qu'il existe une DF $X \to Y$ dans F qui viole la 3FN
2. Créer une relation avec les attributs $X \cup Y$ uniquement
3. Créer une relation avec tous les attributs de R, sauf ceux de Y
4. Appliquer récursivement le principe sur les deux relations créées

Cet algorithme est SPI mais mas forcément SPD.

Exemple :

R(nom,prénom,e-mail,cours,note,vol,ratt)

F = {(nom,prénom $\to$ e-mail) ; (cours,nom,prénom $\to$ note) ; (e-mail $\to$ nom,prénom) ; (cours $\to$ vol) ; (note $\to$ ratt)}

Algorithme de synthèse

Le principe de l'algorithme est (encore plus) simple :

1. Calculer une couverture irredondante irr(F)
2. Partitionner les DF de irr(F) en regroupant dans la même classe celles qui ont la même partie gauche
3. Créer une relation par classe d'équivalence (elles sont toutes au moins 3FN)

A ce stade, **l'algorithme est SPD, mais pas forcément SPI**

4. Si aucune de ces relations ne contient de clef de R, créer une relation supplémentaire avec une clef de R (n'importe laquelle).

Si les 4 étapes sont correctement exécutées, **l'algorithme est SPD et SPI**.

Exemple :

R(nom,prénom,e-mail,cours,note,vol,ratt)

$F = \{$(nom,prénom $\rightarrow$ e-mail) ; (cours,nom,prénom $\rightarrow$ note) ; (e-mail $\rightarrow$ nom,prénom) ; (cours $\rightarrow$ vol) ; (note $\rightarrow$ ratt)$\}$

Faut-il toujours normaliser ?

On peut se dire que dénormaliser un schéma relationnel peut accélérer les calculs (moins de jointures), mais qu'en est-il vraiment ?

Prenons le cas d'une application de commerce dont les tables sont les suivantes :

- Commandes(<u>nocomm</u>, codeclient, …) : 830 commandes enregistrées
- Détailcomm(<u>nocomm</u>,<u>refprod</u>,…) : 2155 lignes de commandes (~2.6 lignes par commande)
- Produits(<u>refprod</u>,nocat,…) : 77 produits (chaque produit est commandé ~28 fois)

Dénormaliser ce schéma (créer une seule relation contenant tout) introduit des redondances (infos sur les commandes et produits) qui peuvent provoquer des anomalies et va prendre plus de place.

Mais combien de place ?

SGBD	Schéma normalisé	Schéma dénormalisé	Variation
MySQL	256	496	+94%
Oracle	133	378	+184%

Faut-il toujours normaliser ?

Prenons les deux requêtes suivantes :

- R_1 : Renvoie le nombre de produits de la catégorie 2 par commande.
- R_2 : Renvoie le nombre de commandes.

Va-t-on vraiment gagner en performance ?

SGBD	Requête	Normalisé	Dénormalisé	Variation
MySQL	R_1	139 397.8743	220.849	-99.8%
	R_2	85.249	220.849	+159.1%
Oracle	R_1	15	20	+33.3%
	R_2	7	19	+171.4%

Pas toujours !

Un bon schéma physique est-il plus intéressant ?

SGBD	Requête	Normalisé	Schéma Physique	Variation
MySQL	R_1	139 397.8743	248.462371	-99.8%
	R_2	85.249	90.141857	+5.4%
Oracle	R_1	15	8	-46.7%
	R_2	7	2	-71.4%

En général oui !

Vision globale du cours

- Introduction SGBD et modèles de données ✓
- Bases de données relationnelles ✓
  - Modèle relationnel ✓
  - Algèbre relationnelle ✓
  - SQL ✓
- Transactions ✓
- Conception de bases de données
  - Dépendances fonctionnelles et normalisation ✓
  - Analyse statique ✘
  - Modèle Entité-Associations ✘
  - Traduction en relationnel ✘

Ce qu'il faut retenir

- Définition d'une dépendance fonctionnelle
- Notions de conséquence logique, de fermeture transitive d'un ensemble, de couverture irredondante
- Définition des formes normales
- Notions de décomposition sans perte d'informations et sans perte de dépendance fonctionnelle
- Algorithmes de décomposition et de synthèse
- Importance de la normalisation