Module BASES DE DONNEES RELATIONNELLES
TD 6
Normalisation

Hervé Owsinski
Herve.Owsinski@uphf.fr

Semestre 3





2026-2027

Rappels

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations “bien conçues”.

Vocabulaire :

- **clé candidate** : ensemble minimal d’attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n’appartiennent à aucune clé candidate.

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations “bien conçues”.

Vocabulaire :

- **clé candidate** : ensemble minimal d’attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n’appartiennent à aucune clé candidate.
- **superclé/clé** : ensemble d’attrinuts qui contient une **clé candidate**.
  Exemple : { **A**, **B**, **E** } noté **ABE** en notation allégée.

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations “bien conçues”.

Vocabulaire :

- **clé candidate** : ensemble minimal d’attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n’appartiennent à aucune clé candidate.
- **superclé/clé** : ensemble d’attrinuts qui contient une **clé candidate**.
  Exemple : { **A**, **B**, **E** } noté **ABE** en notation allégée.

- **1NF** : attributs atomiques et il existe une clé candidate.

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations “bien conçues”.

Vocabulaire :

- **clé candidate** : ensemble minimal d'attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n'appartiennent à aucune clé candidate.
- **superclé/clé** : ensemble d'attrinuts qui contient une **clé candidate**.
  Exemple : { **A**, **B**, **E** } noté **ABE** en notation allégée.

- **1NF** : attributs atomiques et il existe une clé candidate.
- **2NF** : 1NF et pas de DF où **”bout de clé candidate”** $\rightarrow$ **”attr. non clé”**
  ”Le déterminant ne doit pas être composé uniquement d'un bout de clé”.

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations “bien conçues”.

Vocabulaire :

- **clé candidate** : ensemble minimal d'attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n'appartiennent à aucune clé candidate.
- **superclé/clé** : ensemble d'attrinuts qui contient une **clé candidate**.
  Exemple : { **A**, **B**, **E** } noté **ABE** en notation allégée.

- **1NF** : attributs atomiques et il existe une clé candidate.
- **2NF** : 1NF et pas de DF où **”bout de clé candidate”** $\rightarrow$ **”attr. non clé”**
  ”Le déterminant ne doit pas être composé uniquement d'un bout de clé”.
- **3NF** : 2NF et pas de DF où **”attr. non clé”** $\rightarrow$ **”attr. non clé”**.
  ”Le déterminant ne doit pas être composé uniquement de non-clé”.

(Rappel) DF 2NF-compatible ou pas ?

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Considérons **R(A, B, C, D, E)**, attributs clé candidate en rouge, non-clé en bleu.

(DF_1)	**AB**	$\rightarrow$ **C**	2NF-compatible (toute la clé à gauche)
(DF_2)	**A**	$\rightarrow$ **D**	2NF-incompatible (uniquement un bout de clé à gauche)
(DF_3)	**AB**	$\rightarrow$ **E**	2NF-compatible (toute la clé à gauche)
(DF_4)	**C**	$\rightarrow$ **D**	2NF compatible (pas d'attribut clé à gauche)
(DF_5)	**BE**	$\rightarrow$ **C**	2NF compatible car B^+ ne contient pas E
			: on garde **BE** $\rightarrow$ **C**, pas uniquement un bout de clé B.
(DF_6)	**AD**	$\rightarrow$ **E**	2NF incompatible car A^+ contient D donc
			: réductible à **A** $\rightarrow$ **E**, uniquement un bout de clé A

Fermetures :

$$A^+ = \{ A \}^+ = \{ A, \quad D, \quad , E \} = ACDE$$

$$B^+ = \{ B \}^+ = \{ B \} = B$$

$$AB^+ = \{ A, B \}^+ = \{ A, B, \quad C, E, D \} = ABCDE$$

(Rappel) DF 3NF-compatible ou pas ?

3NF : 2NF et pas de DF où **"attr. non clé"** $\rightarrow$ **"attr. non clé"**

Considérons **R(A, B, C, D, E)**, attributs clé candidate en rouge, non-clé en bleu.

(DF_1)	**AB**	$\rightarrow$ **C**	2NF, 3NF compatible (pas de non-clé à gauche)
(DF_2)	**A**	$\rightarrow$ **D**	2NF incompatible donc 3NF incompatible
(DF_3)	**AB**	$\rightarrow$ **E**	2NF, 3NF compatible (pas de non-clé à gauche)
(DF_4)	**C**	$\rightarrow$ **D**	2NF, mais 3NF incompatible (uniquement non-clé à gauche)
(DF_5)	**BE**	$\rightarrow$ **C**	2NF, 3NF compatible (pas que non-clé à gauche)
(DF_6)	**AD**	$\rightarrow$ **E**	2NF incompatible donc 3NF incompatible

Fermetures :

$$A^+ = \{ A \}^+ = \{ A, \quad D, \quad , E \} = ADE$$

$$B^+ = \{ B \}^+ = \{ B \} = B$$

$$AB^+ = \{ A, B \}^+ = \{ A, B, \quad C, E, D \} = ABCDE$$

(Rappel) Méthodologie : couverture minimale

Méthodologie pour trouver une couverture minimale F_{min} :

1. Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).
2. Simplifier les déterminants **réductibles** ($AX \rightarrow C$ avec $A \rightarrow X$ donne $A \rightarrow C$).
3. Utiliser les règles d'Armstrong pour supprimer les DF qu'on peut retrouver en utilisant les autres DF et les règles d'Armstrong.

Astuce : surveillez les attributs recevant plusieurs arcs.

Règles d'Armstrong

R_1	Axiome de réflexivité	si	$Y \subseteq X$	alors	$X \rightarrow Y$
R_2	Axiome d'augmentation	si	$X \rightarrow Y$	alors	$XZ \rightarrow YZ$
R_3	Axiome de transitivité	si	$X \rightarrow Y$ et $Y \rightarrow Z$	alors	$X \rightarrow Z$
R_4	Règle de l'union	si	$X \rightarrow Y$ et $X \rightarrow Z$	alors	$X \rightarrow YZ$
R_5	Règle de la décomposition	si	$X \rightarrow YZ$	alors	$X \rightarrow Y$ et $X \rightarrow Z$
R_6	Règle de la composition	si	$X \rightarrow Y$ et $A \rightarrow B$	alors	$XA \rightarrow YB$
R_7	Règle de pseudo-transitivité	si	$X \rightarrow Y$ et $YZ \rightarrow W$	alors	$XZ \rightarrow W$

Exo 1

Exo 1

Considérons la relation **R(Id_Cours, Id_Etudiant, Age, Note)**
avec les dépendances fonctionnelles :

Id_Cours, Id_Etudiant $\rightarrow$ **Note**
Id_Etudiant $\rightarrow$ **Age**

1. Donnez quelques exemples de tuples illustrant cette relation. (Par exemple plusieurs étudiants suivant plusieurs cours, avec leurs notes et âges.)
2. Déterminez les clés candidates de R.
3. Identifiez les anomalies et redondances possibles (insertion, mise à jour, suppression).
4. Proposez une décomposition de R qui supprime ces anomalies. (Indiquez les relations obtenues et leurs DF correspondantes.)
5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Exo 1

R(Id_Cours, Id_Etudiant, Age, Note)
Id_Cours, Id_Etudiant → Note
Id_Etudiant → Age

1. Donnez quelques exemples de tuples illustrant cette relation. (Par exemple plusieurs étudiants suivant plusieurs cours, avec leurs notes et âges.)

Exo 1

R(Id_Cours, Id_Etudiant, Age, Note)
Id_Cours, Id_Etudiant $\rightarrow$ Note
Id_Etudiant $\rightarrow$ Age

1. Donnez quelques exemples de tuples illustrant cette relation. (Par exemple plusieurs étudiants suivant plusieurs cours, avec leurs notes et âges.)

```
(1, 5, 20, 8),
(2, 5, 20, 16),
(3, 5, 20, 12),
(1, 3, 19, 13),
(2, 3, 19, 11)
```

Exo 1

R(Id_Cours, Id_Etudiant, Age, Note)
Id_Cours, Id_Etudiant → Note
Id_Etudiant → Age

2. Déterminez les clés candidates de R.

Exo 1

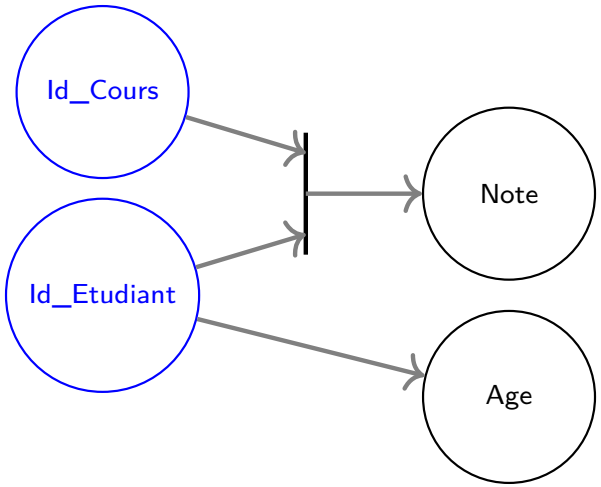
R(Id_Cours, Id_Etudiant, Age, Note)
Id_Cours, Id_Etudiant → Note
Id_Etudiant → Age

2. Déterminez les clés candidates de R.

Les attributs d'une clé candidate apparaissent nécessairement à gauche des DF.

Id_Cours
Id_Etudiant

La connaissance de ce couple permet de déterminer les autres attributs.



Exo 1

R(Id_Cours, Id_Etudiant, Age, Note)
Id_Cours, Id_Etudiant → Note
Id_Etudiant → Age

3. Identifiez les anomalies et redondances possibles (insertion, mise à jour, suppression)

Exo 1

R(Id_Cours, Id_Etudiant, Age, Note)
Id_Cours, Id_Etudiant → Note
Id_Etudiant → Age

3. Identifiez les anomalies et redondances possibles (insertion, mise à jour, suppression)
   - Insertion d'un étudiant déjà présent mais avec un age différent ?
   - Modifier l'age d'un étudiant sur un seul examen et oublier les autres lignes...
   - Supprimer toutes les apparitions d'un étudiant aux examens, supprime aussi la connaissance de son age !

Remarque : hors du cadre "introduction aux BDD", mettre un age est une très mauvaise idée. Pourquoi ?

Exo 1

R(Id_Cours, Id_Etudiant, Age, Note)
Id_Cours, Id_Etudiant → Note
Id_Etudiant → Age

4. Proposez une décomposition de R qui supprime ces anomalies. (Indiquez les relations obtenues et leurs DF correspondantes.)

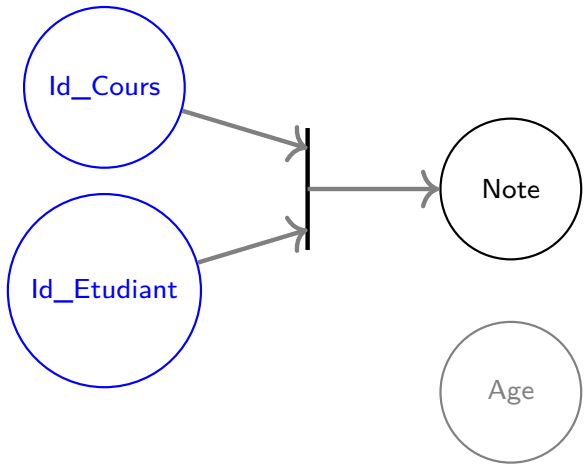
Exo 1

R(**Id_Cours**, **Id_Etudiant**, **Age**, **Note**)
Id_Cours, **Id_Etudiant** $\rightarrow$ **Note** (DF1)
Id_Etudiant $\rightarrow$ **Age** (DF2)

4. Proposez une décomposition de R qui supprime ces anomalies. (Indiquez les relations obtenues et leurs DF correspondantes.)

DF2 pour créer **R2**(**Id_Etudiant**, **Age**)
Après "disparition" DF2, **Age** est isolé dans le graphe : on peut le supprimer de **R**.
On obtient alors :

R(**Id_Cours**, **Id_Etudiant**, **Note**)



On intégre ensuite DF1 directement dans **R**. Avec renommage et FK :

R2 devient **ETUDIANT**(**Id_Etudiant**, **Age**)
R devient **EXAMEN**(**Id_Cours**, **Id_Etudiant**, **Note**)

avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

Exo 1

ETUDIANT R1(**Id_Etudiant**, **Age**)

EXAMEN R2(**Id_Cours**, **#Id_Etudiant**, **Note**)

avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Exo 1

ETUDIANT R1(**Id_Etudiant**, **Age**)

EXAMEN R2(**Id_Cours**, **#Id_Etudiant**, **Note**)

avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Exo 1

ETUDIANT R1(**Id_Etudiant**, **Age**)

EXAMEN R2(**Id_Cours**, **#Id_Etudiant**, **Note**)

avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Sans perte de données :
(cours) $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$

$(R_1 \cap R_2) = \{Id_Etudiant\}$

Exo 1

ETUDIANT R1(**Id_Etudiant**, **Age**)

EXAMEN R2(**Id_Cours**, **#Id_Etudiant**, **Note**)

avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Sans perte de données :
(cours) $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$

$(R_1 \cap R_2) = \{Id_Etudiant\}$

$(R_1 - R_2) = \{\ Age\ \}$

$(R_2 - R_1) = \{\ Id_Cours,\ Note\ \}$

Exo 1

ETUDIANT R1(Id_Etudiant, Age)
EXAMEN R2(Id_Cours, #Id_Etudiant, Note)
avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Sans perte de données :
(cours) $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$

$(R_1 \cap R_2) = \{Id_Etudiant\}$

$(R_1 - R_2) = \{\ Age\ \}$

$(R_2 - R_1) = \{\ Id_Cours,\ Note\ \}$

Or, DF2 ($Id_Etudiant \rightarrow Age$) correspond à $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$.
Conclusion : transformation sans perte de données.

Exo 1

ETUDIANT R1(**Id_Etudiant**, **Age**)

EXAMEN R2(**Id_Cours**, **#Id_Etudiant**, **Note**)

avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Exo 1

ETUDIANT R1(**Id_Etudiant**, **Age**)
EXAMEN R2(**Id_Cours**, **#Id_Etudiant**, **Note**)
avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Sans perte de dépendances :
(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

Nous avions $F = \{ \ Id_Cours, Id_Etudiant \rightarrow Note, \ \ Id_Etudiant \rightarrow Age \ \}$

Exo 1

ETUDIANT R1(**Id_Etudiant**, **Age**)
EXAMEN R2(**Id_Cours**, **#Id_Etudiant**, **Note**)
avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Sans perte de dépendances :
(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

Nous avions $F = \{ \ Id_Cours, Id_Etudiant \rightarrow Note, \ Id_Etudiant \rightarrow Age \ \}$

$\Pi_{R_1}(F) = \{ \ Id_Etudiant \rightarrow Age \ \}$

Exo 1

ETUDIANT R1(**Id_Etudiant**, **Age**)
EXAMEN R2(**Id_Cours**, **#Id_Etudiant**, **Note**)
avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Sans perte de dépendances :
(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

Nous avions $F = \{\ Id_Cours, Id_Etudiant \rightarrow Note,\ \ Id_Etudiant \rightarrow Age\ \}$

$\Pi_{R_1}(F) = \{\ Id_Etudiant \rightarrow Age\ \}$

$\Pi_{R_2}(F) = \{\ Id_Cours, Id_Etudiant \rightarrow Note\ \}$

Exo 1

ETUDIANT R1(**Id_Etudiant**, **Age**)
EXAMEN R2(**Id_Cours**, **#Id_Etudiant**, **Note**)
avec FK(Id_Etudiant) réf. ETUDIANT(Id_Etudiant)

5. Vérifiez que la décomposition est sans perte de données et sans perte de dépendances.

Sans perte de dépendances :
(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

Nous avions $F = \{ \ Id_Cours, Id_Etudiant \rightarrow Note, \ \ Id_Etudiant \rightarrow Age \ \}$

$\Pi_{R_1}(F) = \{ \ Id_Etudiant \rightarrow Age \ \}$

$\Pi_{R_2}(F) = \{ \ Id_Cours, Id_Etudiant \rightarrow Note \ \}$

Nous avons bien $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) \models F$
Conclusion : la transformation conserve les dépendances.

Exo 2

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations “bien conçues”.

Vocabulaire :

- **clé candidate** : ensemble minimal d’attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n’appartiennent à aucune clé candidate.

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations “bien conçues”.

Vocabulaire :

- **clé candidate** : ensemble minimal d’attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n’appartiennent à aucune clé candidate.
- **superclé/clé** : ensemble d’attrinuts qui contient une **clé candidate**.
  Exemple : { **A**, **B**, **E** } noté **ABE** en notation allégée.

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations “bien conçues”.

Vocabulaire :

- **clé candidate** : ensemble minimal d’attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n’appartiennent à aucune clé candidate.
- **superclé/clé** : ensemble d’attrinuts qui contient une **clé candidate**.
  Exemple : { **A**, **B**, **E** } noté **ABE** en notation allégée.

- **1NF** : attributs atomiques et il existe une clé candidate.

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations “bien conçues”.

Vocabulaire :

- **clé candidate** : ensemble minimal d'attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n'appartiennent à aucune clé candidate.
- **superclé/clé** : ensemble d'attrinuts qui contient une **clé candidate**.
  Exemple : { **A**, **B**, **E** } noté **ABE** en notation allégée.

- **1NF** : attributs atomiques et il existe une clé candidate.
- **2NF** : 1NF et pas de DF où **”bout de clé candidate”** $\rightarrow$ **”attr. non clé”**
  ”Le déterminant ne doit pas être composé uniquement d'un bout de clé”.

(Rappel) Formes normales

La théorie de la normalisation est fondée sur deux concepts : DF et FN

- **dépendances fonctionnelles** : traduisent les contraintes sur les données.
- **formes normales** : définissent des familles de relations "bien conçues".

Vocabulaire :

- **clé candidate** : ensemble minimal d'attr permettant de déterminer les autres.
  Exemple : { **A**, **B** } noté **AB** en notation allégée.
- Un attr. **non clé** est un attribut qui n'appartiennent à aucune clé candidate.
- **superclé/clé** : ensemble d'attrinuts qui contient une **clé candidate**.
  Exemple : { **A**, **B**, **E** } noté **ABE** en notation allégée.

- **1NF** : attributs atomiques et il existe une clé candidate.
- **2NF** : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**
  "Le déterminant ne doit pas être composé uniquement d'un bout de clé".
- **3NF** : 2NF et pas de DF où **"attr. non clé"** $\rightarrow$ **"attr. non clé"**.
  "Le déterminant ne doit pas être composé uniquement de non-clé".

(Rappel) DF 2NF-compatible ou pas ?

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Considérons **R(A, B, C, D, E)**, attributs clé candidate en rouge, non-clé en bleu.

(DF_1)	**AB**	$\rightarrow$ **C**	2NF-compatible (toute la clé à gauche)
(DF_2)	**A**	$\rightarrow$ **D**	2NF-incompatible (uniquement un bout de clé à gauche)
(DF_3)	**AB**	$\rightarrow$ **E**	2NF-compatible (toute la clé à gauche)
(DF_4)	**C**	$\rightarrow$ **D**	2NF compatible (pas d'attribut clé à gauche)
(DF_5)	**BE**	$\rightarrow$ **C**	2NF compatible car B^+ ne contient pas E
			: on garde **BE** $\rightarrow$ **C**, pas uniquement un bout de clé B.
(DF_6)	**AD**	$\rightarrow$ **E**	2NF incompatible car A^+ contient D donc
			: réductible à **A** $\rightarrow$ **E**, uniquement un bout de clé A

Fermetures :

$$A^+ = \{ A \}^+ = \{ A, \quad D, \quad , E \} = ACDE$$

$$B^+ = \{ B \}^+ = \{ B \} = B$$

$$AB^+ = \{ A, B \}^+ = \{ A, B, \quad C, E, D \} = ABCDE$$

(Rappel) DF 3NF-compatible ou pas ?

3NF : 2NF et pas de DF où **"attr. non clé" → "attr. non clé"**

Considérons **R(A, B, C, D, E)**, attributs clé candidate en rouge, non-clé en bleu.

(DF_1)	**AB**	$\rightarrow$ **C**	2NF, 3NF compatible (pas de non-clé à gauche)
(DF_2)	**A**	$\rightarrow$ **D**	2NF incompatible donc 3NF incompatible
(DF_3)	**AB**	$\rightarrow$ **E**	2NF, 3NF compatible (pas de non-clé à gauche)
(DF_4)	**C**	$\rightarrow$ **D**	2NF, mais 3NF incompatible (uniquement non-clé à gauche)
(DF_5)	**BE**	$\rightarrow$ **C**	2NF, 3NF compatible (pas que non-clé à gauche)
(DF_6)	**AD**	$\rightarrow$ **E**	2NF incompatible donc 3NF incompatible

Fermetures :

$$A^+ = \{ A \}^+ = \{ A, \quad D, \quad , E \} = ADE$$

$$B^+ = \{ B \}^+ = \{ B \} = B$$

$$AB^+ = \{ A, B \}^+ = \{ A, B, \quad C, E, D \} = ABCDE$$

Exo 2

On considère le schéma relationnel

$$\mathbf{R}(\mathbf{Cours}, \mathbf{Prof}, \mathbf{Heure}, \mathbf{Salle}, \mathbf{Etudiant}, \mathbf{Note})$$

avec les dépendances fonctionnelles suivantes :

$\mathbf{Cours} \rightarrow \mathbf{Prof}$

$\mathbf{Heure}, \mathbf{Prof} \rightarrow \mathbf{Salle}$

$\mathbf{Heure}, \mathbf{Etudiant} \rightarrow \mathbf{Salle}$

$\mathbf{Heure}, \mathbf{Salle} \rightarrow \mathbf{Cours}$

$\mathbf{Cours}, \mathbf{Etudiant} \rightarrow \mathbf{Note}$

$\mathbf{Heure}, \mathbf{Salle}, \mathbf{Etudiant} \rightarrow \mathbf{Note}$

1. Donner la clé de cette relation
2. Ce schéma est-il en 2NF ?
3. Pourquoi ce schéma n'est-il pas en 3NF ?
4. Donner ici une décomposition en 3NF. (Préciser comment elle est obtenue)
5. La décomposition préserve-t-elle les dépendances fonctionnelles ?
6. Y a-t-il perte par jointure ? Sinon modifier le schéma pour qu'il n'y ait pas de perte par jointure.

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**

(DF2) **Heure, Prof** $\rightarrow$ **Salle**

(DF3) **Heure, Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure, Salle** $\rightarrow$ **Cours**

(DF5) **Cours, Etudiant** $\rightarrow$ **Note**

(DF6) **Heure, Salle, Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Heure et **Etudiant** jamais à droite.
Les attributs de la clé comportent nécessairement { **Heure**, **Etudiant** }
Est-ce suffisant pour déterminer les 4 autres ?

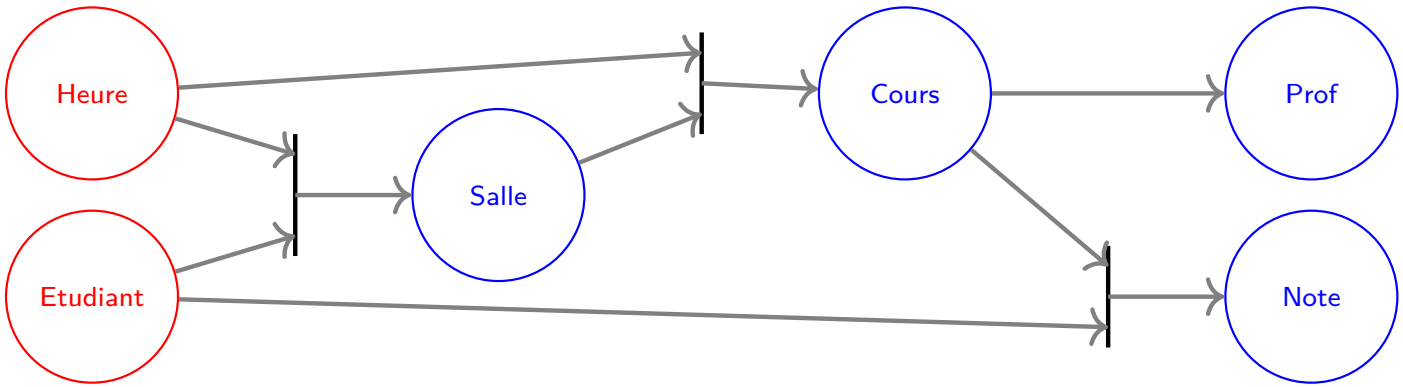
Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Heure et **Etudiant** jamais à droite.
Les attributs de la clé comportent nécessairement { **Heure**, **Etudiant** }
Est-ce suffisant pour déterminer les 4 autres ?



Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Démonstration :(

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Démonstration :(

Nous supposons que {**Heure**, **Etudiant**} est la clé candidate.
Ces deux éléments doivent être (1) minimaux et (2) permettre de trouver les autres.
(1) Nécessairement minimaux car jamais présents à droite.
(2) La fermeture {**Heure**, **Etudiant**}$^+$ doit être égale à l'ensemble des attributs.

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Démonstration :(

Nous supposons que {**Heure**, **Etudiant**} est la clé candidate.
Ces deux éléments doivent être (1) minimaux et (2) permettre de trouver les autres.
(1) Nécessairement minimaux car jamais présents à droite.
(2) La fermeture {**Heure**, **Etudiant**}$^+$ doit être égale à l'ensemble des attributs.
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle** } avec DF3

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Démonstration :(

Nous supposons que {**Heure**, **Etudiant**} est la clé candidate.
Ces deux éléments doivent être (1) minimaux et (2) permettre de trouver les autres.
(1) Nécessairement minimaux car jamais présents à droite.
(2) La fermeture {**Heure**, **Etudiant**}$^+$ doit être égale à l'ensemble des attributs.
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle** } avec DF3
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle**, **Cours** } avec DF4

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Démonstration :(

Nous supposons que {**Heure**, **Etudiant**} est la clé candidate.
Ces deux éléments doivent être (1) minimaux et (2) permettre de trouver les autres.
(1) Nécessairement minimaux car jamais présents à droite.
(2) La fermeture {**Heure**, **Etudiant**}$^+$ doit être égale à l'ensemble des attributs.
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle** } avec DF3
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle**, **Cours** } avec DF4
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle**, **Cours**, **Note** } avec DF5 ou DF6

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Démonstration :(

Nous supposons que {**Heure**, **Etudiant**} est la clé candidate.
Ces deux éléments doivent être (1) minimaux et (2) permettre de trouver les autres.
(1) Nécessairement minimaux car jamais présents à droite.
(2) La fermeture {**Heure**, **Etudiant**}$^+$ doit être égale à l'ensemble des attributs.
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle** } avec DF3
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle**, **Cours** } avec DF4
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle**, **Cours**, **Note** } avec DF5 ou DF6
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle**, **Cours**, **Note**, **Prof** } avec DF1

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

1. Donner la clé de cette relation.

Démonstration :(

Nous supposons que {**Heure**, **Etudiant**} est la clé candidate.
Ces deux éléments doivent être (1) minimaux et (2) permettre de trouver les autres.
(1) Nécessairement minimaux car jamais présents à droite.
(2) La fermeture {**Heure**, **Etudiant**}$^+$ doit être égale à l'ensemble des attributs.
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle** } avec DF3
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle**, **Cours** } avec DF4
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle**, **Cours**, **Note** } avec DF5 ou DF6
{**Heure**, **Etudiant**}$^+$ = { **Heure**, **Etudiant**, **Salle**, **Cours**, **Note**, **Prof** } avec DF1
(1) + (2) : il s'agit bien d'une clé candidate.
:)

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

2. Ce schéma est-il en 2NF ?

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

2. Ce schéma est-il en 2NF ?

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

2. Ce schéma est-il en 2NF ?

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Démonstration :(

(DF1) 2NF compatible (aucun attribut clé à gauche)

:)

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

2. Ce schéma est-il en 2NF ?

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Démonstration :(

(DF1) 2NF compatible (aucun attribut clé à gauche)

(DF2) 2NF compatible ($H^+ = H$, on garde**HP** à gauche, pas juste un bout de clé)

:)

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

2. Ce schéma est-il en 2NF ?

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Démonstration :(

(DF1) 2NF compatible (aucun attribut clé à gauche)

(DF2) 2NF compatible ($H^+ = H$, on garde**HP** à gauche, pas juste un bout de clé)

(DF3) 2NF compatible (**clé HE**)

:)

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

2. Ce schéma est-il en 2NF ?

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Démonstration :(

(DF1) 2NF compatible (aucun attribut clé à gauche)

(DF2) 2NF compatible ($H^+ = H$, on garde**HP** à gauche, pas juste un bout de clé)

(DF3) 2NF compatible (**clé HE**)

(DF4) 2NF compatible ($H^+ = H$, on garde **HS** à gauche, pas juste un bout de clé)

:)

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

2. Ce schéma est-il en 2NF ?

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Démonstration :(

(DF1) 2NF compatible (aucun attribut clé à gauche)

(DF2) 2NF compatible ($H^+ = H$, on garde**HP** à gauche, pas juste un bout de clé)

(DF3) 2NF compatible (**clé HE**)

(DF4) 2NF compatible ($H^+ = H$, on garde **HS** à gauche, pas juste un bout de clé)

(DF5) 2NF compatible ($E^+ = E$, on garde **CE** à gauche, pas juste un bout de clé)

:)

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

2. Ce schéma est-il en 2NF ?

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Démonstration :(

(DF1) 2NF compatible (aucun attribut clé à gauche)

(DF2) 2NF compatible ($H^+ = H$, on garde**HP** à gauche, pas juste un bout de clé)

(DF3) 2NF compatible (**clé HE**)

(DF4) 2NF compatible ($H^+ = H$, on garde **HS** à gauche, pas juste un bout de clé)

(DF5) 2NF compatible ($E^+ = E$, on garde **CE** à gauche, pas juste un bout de clé)

(DF6) 2NF compatible (toute la **clé HE** à gauche).

:)

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

3. Pourquoi ce schéma n'est-il pas en 3NF ?

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

3. Pourquoi ce schéma n'est-il pas en 3NF ?

3NF : 2NF et pas de DF où **"attr. non clé"** $\rightarrow$ **"attr. non clé"**.

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure, Prof** $\rightarrow$ **Salle**
(DF3) **Heure, Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure, Salle** $\rightarrow$ **Cours**
(DF5) **Cours, Etudiant** $\rightarrow$ **Note**
(DF6) **Heure, Salle, Etudiant** $\rightarrow$ **Note**

3. Pourquoi ce schéma n'est-il pas en 3NF ?

3NF : 2NF et pas de DF où **"attr. non clé"** $\rightarrow$ **"attr. non clé"**.
La clé est **{ Heure, Etudiant }** . Les autres attributs sont non clés.
Cette relation n'est pas en 3NF.

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** → **Prof**
(DF2) **Heure**, **Prof** → **Salle**
(DF3) **Heure**, **Etudiant** → **Salle**
(DF4) **Heure**, **Salle** → **Cours**
(DF5) **Cours**, **Etudiant** → **Note**
(DF6) **Heure**, **Salle**, **Etudiant** → **Note**

3. Pourquoi ce schéma n'est-il pas en 3NF ?

3NF : 2NF et pas de DF où **"attr. non clé"** → **"attr. non clé"**.
La clé est **{ Heure, Etudiant }** . Les autres attributs sont non clés.
Cette relation n'est pas en 3NF.

Démonstration **:(** deux façons différentes de le démontrer
a) La relation n'était pas en 2NF condition indispensable.
b) (DF 1) : **Prof** dépend uniquement de **Cours**, non clé.
:)

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

4. Donner ici une décomposition en 3NF. (Préciser comment elle est obtenue)

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

4. Donner ici une décomposition en 3NF. (Préciser comment elle est obtenue)

3NF : 2NF et pas de DF où **"attr. non clé"** $\rightarrow$ **"attr. non clé"**.
Seule DF1 posait problème. Il va la faire "disparaître" de **R**.

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) **Cours** $\rightarrow$ **Prof**
(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**
(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**
(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**
(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**
(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

4. Donner ici une décomposition en 3NF. (Préciser comment elle est obtenue)

Il faut faire disparaître DF1 en l'intégrant dans une relation.

Exo 2

R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF1) Cours → Prof
(DF2) Heure, Prof → Salle
(DF3) Heure, Etudiant → Salle
(DF4) Heure, Salle → Cours
(DF5) Cours, Etudiant → Note
(DF6) Heure, Salle, Etudiant → Note

4. Donner ici une décomposition en 3NF. (Préciser comment elle est obtenue)

Il faut faire disparaître DF1 en l'intégrant dans une relation.

R2(Cours, Prof)
R(Cours, Prof, Heure, Salle, Etudiant, Note)

(DF2) Heure, Prof → Salle
(DF3) Heure, Etudiant → Salle
(DF4) Heure, Salle → Cours
(DF5) Cours, Etudiant → Note
(DF6) Heure, Salle, Etudiant → Note

Exo 2

R2(**Cours**, **Prof**)

R(**Cours**, **Prof**, **Heure**, **Salle**, **Etudiant**, **Note**)

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

5. La décomposition préserve-t-elle les dépendances fonctionnelles ?

Exo 2

R2(**Cours**, **Prof**)

R(**Cours**, **Prof**, **Heure**, **Salle**, **Etudiant**, **Note**)

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

5. La décomposition préserve-t-elle les dépendances fonctionnelles ?

Sans perte de dépendances :

(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

avec

(DF2) à (DF6) mais également :

(DF1) **Cours** $\rightarrow$ **Prof**

Exo 2

R2(**Cours**, **Prof**)

R(**Cours**, **Prof**, **Heure**, **Salle**, **Etudiant**, **Note**)

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

5. La décomposition préserve-t-elle les dépendances fonctionnelles ?

Sans perte de dépendances :

(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

avec

(DF2) à (DF6) mais également :

(DF1) **Cours** $\rightarrow$ **Prof**

Démonstration :(

(DF 1) s'obtient directement à partir de la projection sur R2.

Les autres DF sont issues de R.

Leur union donne bien entendu l'ensemble des DF 1 à 6.

:)

Exo 2

R2(**Cours**, **Prof**)

R(**Cours**, **Prof**, **Heure**, **Salle**, **Etudiant**, **Note**)

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

6. Y a-t-il perte par jointure ? Sinon modifier le schéma pour qu'il n'y ait pas de perte par jointure.

Exo 2

R2(**Cours**, **Prof**)

R(**Cours**, **Prof**, **Heure**, **Salle**, **Etudiant**, **Note**)

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

6. Y a-t-il perte par jointure ? Sinon modifier le schéma pour qu'il n'y ait pas de perte par jointure.

Sans perte de données (disparition ou ajout)
(cours) $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$
Démonstration :(

Exo 2

R2(**Cours**, **Prof**)

R(**Cours**, **Prof**, **Heure**, **Salle**, **Etudiant**, **Note**)

(DF2) **Heure**, **Prof** $\rightarrow$ **Salle**

(DF3) **Heure**, **Etudiant** $\rightarrow$ **Salle**

(DF4) **Heure**, **Salle** $\rightarrow$ **Cours**

(DF5) **Cours**, **Etudiant** $\rightarrow$ **Note**

(DF6) **Heure**, **Salle**, **Etudiant** $\rightarrow$ **Note**

6. Y a-t-il perte par jointure ? Sinon modifier le schéma pour qu'il n'y ait pas de perte par jointure.

Sans perte de données (disparition ou ajout)

(cours) $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$

Démonstration :(

$(R \cap R_2) = \{Cours, Prof\}$

$(R - R_2) = \{Heure, Salle, Etudiant, Note\}$

$(R_2 - R) = \varnothing$

On ne peut donc pas garantir qu'il n'y ai pas de perte de données.

:)

Exo 2

R(C, P, H, S, E, N)

(DF1) C → P
(DF2) H, P → S
(DF3) H, E → S
(DF4) H, S → C
(DF5) C, E → N
(DF6) H, S, E → N

Poursuite de la décomposition

La 3 NF ne garantit pas à elle seule la préservation des données. Il va falloir poursuivre la décomposition en se basant sur une couverture minimale F_{min} de DF telle que $F^{+}_{min} = F^{+}$.
Méthodologie pour trouver une couverture minimale F_{min} :

1. Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).
2. Simplifier les déterminants **réductibles** ($AX \rightarrow C$ avec $A \rightarrow X$ donne $A \rightarrow C$).
3. Utiliser les règles d'Armstrong pour supprimer les DF qu'on peut retrouver en utilisant les autres DF et les règles d'Armstrong.

Astuce : surveillez les attributs recevant plusieurs arcs.

Exo 2

R(C, P, H, S, E, N)

(DF1) C → P

(DF2) H, P → S

(DF3) H, E → S

(DF4) H, S → C

(DF5) C, E → N

(DF6) H, S, E → N

Etape 1

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).

Rien à faire ici.

Exo 2

R(C, P, H, S, E, N)

(DF1) C → P

(DF2) H, P → S

(DF3) H, E → S

(DF4) H, S → C

(DF5) C, E → N

(DF6) H, S, E → N

Etape 2

- Simplifier les déterminants **réductibles** ($AX \rightarrow C$ avec $A \rightarrow X$ donne $A \rightarrow C$).

Rien à faire ici.

Exo 2

R(C, P, H, S, E, N)

(DF1) C → P
(DF2) H, P → S
(DF3) H, E → S
(DF4) H, S → C
(DF5) C, E → N
(DF6) H, S, E → N

Etape 3

- Utiliser les règles d'Armstrong pour supprimer les DF qu'on peut retrouver en utilisant les autres DF et les règles d'Armstrong.

Astuce : surveillez les attributs recevant plusieurs arcs.

Règles d'Armstrong

R_1	Axiome de réflexivité	si	$Y \subseteq X$	alors	$X \to Y$
R_2	Axiome d'augmentation	si	$X \to Y$	alors	$XZ \to YZ$
R_3	Axiome de transitivité	si	$X \to Y$ et $Y \to Z$	alors	$X \to Z$
R_4	Règle de l'union	si	$X \to Y$ et $X \to Z$	alors	$X \to YZ$
R_5	Règle de la décomposition	si	$X \to YZ$	alors	$X \to Y$ et $X \to Z$
R_6	Règle de la composition	si	$X \to Y$ et $A \to B$	alors	$XA \to YB$
R_7	Règle de pseudo-transitivité	si	$X \to Y$ et $YZ \to W$	alors	$XZ \to W$

Exo 2

R(C, P, H, S, E, N)

(DF1) C → P

(DF2) H, P → S

(DF3) H, E → S

(DF4) H, S → C

(DF5) C, E → N

(DF6) H, S, E → N

Etape 3 - Recherche de la couverture minimale

On utilise la pseudo-transitivité sur DP4 et DF 5 :

$$(HS \rightarrow C) \wedge (CE \rightarrow N) \underset{\text{pseudo-transitivité}}{\Longrightarrow} HSE \rightarrow N$$

On peut donc se passer de la DF6 puisqu'on la retrouve en utilisant DF4 et DF5.

On ne peut pas se passer facilement de DF2 ou DF3 à cause de références circulaires.

Exo 2

R(C, P, H, S, E, N)

(DF1) C → P

(DF2) H, P → S

(DF3) H, E → S

(DF4) H, S → C

(DF5) C, E → N

Il reste donc à établir les relations une à une :

Exo 2

R(**C**, **P**, **H**, **S**, **E**, **N**)

(DF1) **C** → **P**

(DF2) **H**, **P** → **S**

(DF3) **H**, **E** → **S**

(DF4) **H**, **S** → **C**

(DF5) C, E → N

Il reste donc à établir les relations une à une :

DF5 donne **R2**(C, E, N) reste **R**(C,P,H,S,E) N isolé.

Exo 2

R(**C**, **P**, **H**, **S**, **E**, **N**)

(DF1) C → P

(DF2) **H**, **P** → **S**

(DF3) **H**, **E** → **S**

(DF4) **H**, **S** → **C**

(DF5) C, E → N

Il reste donc à établir les relations une à une :

DF5 donne **R2**(C, E, N) reste **R**(C,P,H,S,E) N isolé.

DF1 donne **R3**(C, P) reste **R**(C,P,H,S,E)

Exo 2

R(C, P, H, S, E, N)

(DF1) $C \rightarrow P$

(DF2) $H, P \rightarrow S$

(DF3) $H, E \rightarrow S$

(DF4) $H, S \rightarrow C$

(DF5) $C, E \rightarrow N$

Il reste donc à établir les relations une à une :

DF5 donne **R2**(C, E, N) reste **R**(C,P,H,S,E) N isolé.

DF1 donne **R3**(C, P) reste **R**(C,P,H,S,E)

DF2 donne **R4**(H, P, S) reste **R**(C,H,S,E) P isolé de R.

Exo 2

R(C, P, H, S, E, N)

(DF1) $C \rightarrow P$

(DF2) $H, P \rightarrow S$

(DF3) $H, E \rightarrow S$

(DF4) $H, S \rightarrow C$

(DF5) $C, E \rightarrow N$

Il reste donc à établir les relations une à une :

DF5 donne **R2**(C, E, N) reste **R**(C,P,H,S,E) N isolé.

DF1 donne **R3**(C, P) reste **R**(C,P,H,S,E)

DF2 donne **R4**(H, P, S) reste **R**(C,H,S,E) P isolé de R.

DF4 donne **R5**(H, S, C) reste **R**(H,S,E) C isolé de R.

Il ne reste plus que DF3 qui permet de rajouter la PK de R :

Exo 2

R(**C**, **P**, **H**, **S**, **E**, **N**)

(DF1)	C → P
(DF2)	H, P → S
(DF3)	H, E → S
(DF4)	H, S → C
(DF5)	C, E → N

Il reste donc à établir les relations une à une :

DF5 donne **R2**(C, E, N) reste **R**(C,P,H,S,E) N isolé.

DF1 donne **R3**(C, P) reste **R**(C,P,H,S,E)

DF2 donne **R4**(H, P, S) reste **R**(C,H,S,E) P isolé de R.

DF4 donne **R5**(H, S, C) reste **R**(H,S,E) C isolé de R.

Il ne reste plus que DF3 qui permet de rajouter la PK de R :

DF3 donne **R1**(H, E, S) reste **R**(∅) HES isolés de R.

On remplace donc **R** et ses DF par 6 relations **R1 R2 R3 R4 R5** et des PK/FK.

Exo 3

Exo 3 : (en fin de TD plutôt)

On dispose de la relation suivante :

Video (No_Client, Nom_Client, Prenom_Client, Ville_Client, DateDebut_AbonnementClient, No_Cassette, Titre_Film, PrixAchat_Cassette, Nom_Editeur, Ville_Editeur, Duree_Film, Nationalite_Film, Nom_Realisateur, AnneeSortie_Film, Genre, DateLocation_Cassette, Duree_Location)

Hypothèse : on considère que les titres de film sont uniques (pas de remake...) et on souhaite garder un historique des locations de vidéo.

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)
2. Décomposez la relation jusqu'à la 2NF, tout en préservant les dépendances et en évitant la perte d'information. (Indiquez les nouvelles relations et leurs DF.)
3. Vérifiez si les relations obtenues sont en 3NF. Si non, poursuivez la décomposition et justifiez.

Rappels

- **1NF** : attributs atomiques et il existe une clé candidate.

- **2NF** : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**
  "Le déterminant ne doit pas être composé uniquement d'un bout de clé".

- **3NF** : 2NF et pas de DF où **"attr. non clé"** $\rightarrow$ **"attr. non clé"**.
  "Le déterminant ne doit pas être composé uniquement de non-clé".

Exo 3-1

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Exo 3-1

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Prouver non-2NF ? Il faut trouver la clé et une DF non 2NF-compatible.

Exo 3-1

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Prouver non-2NF ? Il faut trouver la clé et une DF non 2NF-compatible.
Trouver la clé ? Il faut trouver les DF issues de cette relation.

Exo 3-1

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Prouver non-2NF ? Il faut trouver la clé et une DF non 2NF-compatible.
Trouver la clé ? Il faut trouver les DF issues de cette relation.
Quelles sont les DF logiques/évidentes ?

No_Cli $\rightarrow$ Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli

Exo 3-1

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Prouver non-2NF ? Il faut trouver la clé et une DF non 2NF-compatible.
Trouver la clé ? Il faut trouver les DF issues de cette relation.
Quelles sont les DF logiques/évidentes ?

```
No_Cli → Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli
No_Cas → Titre_Film, PrixAchat_Cas, Nom_Edi
```

Exo 3-1

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Prouver non-2NF ? Il faut trouver la clé et une DF non 2NF-compatible.
Trouver la clé ? Il faut trouver les DF issues de cette relation.
Quelles sont les DF logiques/évidentes ?

```
No_Cli → Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli
No_Cas → Titre_Film, PrixAchat_Cas, Nom_Edi
Nom_Edi → Ville_Edi
```

Exo 3-1

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** → **"attr. non clé"**

Prouver non-2NF ? Il faut trouver la clé et une DF non 2NF-compatible.
Trouver la clé ? Il faut trouver les DF issues de cette relation.
Quelles sont les DF logiques/évidentes ?

```
No_Cli → Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli
No_Cas → Titre_Film, PrixAchat_Cas, Nom_Edi
Nom_Edi → Ville_Edi
Titre_Film → Duree_Film, Nat_Film, Nom_Rea, AnnSor_Film, Genre
```

Exo 3-1

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** → **"attr. non clé"**

Prouver non-2NF ? Il faut trouver la clé et une DF non 2NF-compatible.
Trouver la clé ? Il faut trouver les DF issues de cette relation.
Quelles sont les DF logiques/évidentes ?

```
No_Cli → Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli
No_Cas → Titre_Film, PrixAchat_Cas, Nom_Edi
Nom_Edi → Ville_Edi
Titre_Film → Duree_Film, Nat_Film, Nom_Rea, AnnSor_Film, Genre
No_Cas, DateLoc_Cas → No_Cli, Duree_Loc
```

Exo 3-1

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Prouver non-2NF ? Il faut trouver la clé et une DF non 2NF-compatible.
Trouver la clé ? Il faut trouver les DF issues de cette relation.
Quelles sont les DF logiques/évidentes ?

```
No_Cli → Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli
No_Cas → Titre_Film, PrixAchat_Cas, Nom_Edi
Nom_Edi → Ville_Edi
Titre_Film → Duree_Film, Nat_Film, Nom_Rea, AnnSor_Film, Genre
No_Cas, DateLoc_Cas → No_Cli, Duree_Loc
```

(1) {No_Cas, DateLoc_Cas} $\in$ cle car jamais déterminés à droite
(2) Fermeture {No_Cas, DateLoc_Cas}$^+$ donnne bien l'ensemble des attributs.
(3) Minimal car {No_Cas}$^+$ ou {DateLoc_Cas}$^+$ seule ne donne pas les autres.

Exo 3-1

Video (No_Cli, Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli, No_Cas, Titre_Film, PrixAchat_Cas, Nom_Edi, Ville_Edi, Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre, DateLoc_Cas, Duree_Loc)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** → **"attr. non clé"**

Conclusion (1)+(2)+(3) : **{No_Cas, DateLoc_Cas}** est l'unique clé candidate.

(DF01-04) **No_Cli** → **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF05-07) **No_Cas** → **Titre_Film, PrixAchat_Cas, Nom_Edi**
(DF08) **Nom_Edi** → **Ville_Edi**
(DF09-13) **Titre_Film** → **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

Exo 3-1

Video (No_Cli, Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli, No_Cas, Titre_Film, PrixAchat_Cas, Nom_Edi, Ville_Edi, Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre, DateLoc_Cas, Duree_Loc)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** → **"attr. non clé"**

Conclusion (1)+(2)+(3) : **{No_Cas, DateLoc_Cas}** est l'unique clé candidate.

(DF01-04) **No_Cli** → **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF05-07) **No_Cas** → **Titre_Film, PrixAchat_Cas, Nom_Edi**
(DF08) **Nom_Edi** → **Ville_Edi**
(DF09-13) **Titre_Film** → **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

Les **DF5** à **DF7** ne sont pas 2NF-compatibles.

Exo 3-1

Video (No_Cli, Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli, No_Cas, Titre_Film, PrixAchat_Cas, Nom_Edi, Ville_Edi, Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre, DateLoc_Cas, Duree_Loc)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** $\rightarrow$ **"attr. non clé"**

Conclusion (1)+(2)+(3) : **{No_Cas, DateLoc_Cas}** est l'unique clé candidate.

(DF01-04) **No_Cli** $\rightarrow$ **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF05-07) **No_Cas** $\rightarrow$ **Titre_Film, PrixAchat_Cas, Nom_Edi**
(DF08) **Nom_Edi** $\rightarrow$ **Ville_Edi**
(DF09-13) **Titre_Film** $\rightarrow$ **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** $\rightarrow$ **No_Cli, Duree_Loc**

Les **DF5** à **DF7** ne sont pas 2NF-compatibles.
On a des contre-exemples donc la relation **Video** n'est pas en 2NF.

Exo 3-1

Video (No_Cli, Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli, No_Cas, Titre_Film, PrixAchat_Cas, Nom_Edi, Ville_Edi, Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre, DateLoc_Cas, Duree_Loc)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** → **"attr. non clé"**

Conclusion (1)+(2)+(3) : **{No_Cas, DateLoc_Cas}** est l'unique clé candidate.

(DF01-04) **No_Cli** → **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF05-07) **No_Cas** → **Titre_Film, PrixAchat_Cas, Nom_Edi**
(DF08) **Nom_Edi** → **Ville_Edi**
(DF09-13) **Titre_Film** → **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

Les **DF5** à **DF7** ne sont pas 2NF-compatibles.
On a des contre-exemples donc la relation **Video** n'est pas en 2NF.

Lles DF5, DF6 et DF7 sont les seules non-2NF compatibles.

Exo 3-2

2. Décomposez la relation jusqu'à la 2NF, tout en préservant les dépendances et en évitant la perte d'information. (Indiquez les nouvelles relations et leurs DF.)

Exo 3-2

2. Décomposez la relation jusqu'à la 2NF, tout en préservant les dépendances et en évitant la perte d'information. (Indiquez les nouvelles relations et leurs DF.)

(DF01-04) **No_Cli** → **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF05-07) **No_Cas** → **Titre_Film, PrixAchat_Cas, Nom_Edi**
(DF08) **Nom_Edi** → **Ville_Edi**
(DF09-13) **Titre_Film** → **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

On extrait DF5 DF6 et DF7 de **Video** et on en fait une relation.

(DF05-07) **No_Cas** → **Titre_Film , PrixAchat_Cas, Nom_Edi**

Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

Exo 3-2

2. Décomposez la relation jusqu'à la 2NF, tout en préservant les dépendances et en évitant la perte d'information. (Indiquez les nouvelles relations et leurs DF.)

(DF01-04) **No_Cli** → **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**

(DF08) **Nom_Edi** → **Ville_Edi**
(DF09-13) **Titre_Film** → **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

On extrait DF5 DF6 et DF7 de **Video** et on en fait une relation.

(DF05-07) **No_Cas** → **Titre_Film , PrixAchat_Cas, Nom_Edi**
Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

On constate que **PrixAchat_Cas** n'apparaît plus dans les DF qui restent.
Etant isolé dans le graphe, on le supprime de **Video** :

Exo 3-2

2. Décomposez la relation jusqu'à la 2NF, tout en préservant les dépendances et en évitant la perte d'information. (Indiquez les nouvelles relations et leurs DF.)

(DF01-04) **No_Cli** → **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**

(DF08) **Nom_Edi** → **Ville_Edi**
(DF09-13) **Titre_Film** → **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

On extrait DF5 DF6 et DF7 de **Video** et on en fait une relation.

(DF05-07) **No_Cas** → **Titre_Film , PrixAchat_Cas, Nom_Edi**
Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

On constate que **PrixAchat_Cas** n'apparaît plus dans les DF qui restent.
Etant isolé dans le graphe, on le supprime de **Video** :

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **Nom_Edi**, **Ville_Edi**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

Exo 3-2

Sans perte de dépendances :
(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

Démonstration :(

Exo 3-2

Sans perte de dépendances :
(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

Démonstration :(
Reprenons l'ensemble F et les relations :
(DF01-04) `No_Cli` → `Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli`
(DF05-07) `No_Cas` → `Titre_Film, PrixAchat_Cas, Nom_Edi`
(DF08) `Nom_Edi` → `Ville_Edi`
(DF09-13) `Titre_Film` → `Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre`
(DF14-15) `No_Cas, DateLoc_Cas` → `No_Cli, Duree_Loc`

Video (No_Cli, Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli, No_Cas, Titre_Film, Nom_Edi, Ville_Edi, Duree_Film, Nat_Film, Nom_Rea, AnnSor_Film, Genre, DateLoc_Cas, Duree_Loc)

Cassette (No_Cas, Titre_Film, PrixAchat_Cas, Nom_Edi)

Exo 3-2

Sans perte de dépendances :
(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

Démonstration :(
Reprenons l'ensemble F et les relations :

(DF01-04) **No_Cli** → **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF05-07) **No_Cas** → **Titre_Film, PrixAchat_Cas, Nom_Edi**
(DF08) **Nom_Edi** → **Ville_Edi**
(DF09-13) **Titre_Film** → **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **Nom_Edi**, **Ville_Edi**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

$\Pi_{Video}(F) = \{DF_1 ... DF_5, DF_7 ... DF_{15}\}$ mais pas (DF6) car il manque **PrixAchat_Cas** dans **Video**.

Exo 3-2

Sans perte de dépendances :
(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

Démonstration :(
Reprenons l'ensemble F et les relations :
(DF01-04) **No_Cli** → **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF05-07) **No_Cas** → **Titre_Film, PrixAchat_Cas, Nom_Edi**
(DF08) **Nom_Edi** → **Ville_Edi**
(DF09-13) **Titre_Film** → **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **Nom_Edi**, **Ville_Edi**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

$\Pi_{Video}(F) = \{DF_1 ... DF_5, DF_7 ... DF_{15}\}$ mais pas (DF6) car il manque **PrixAchat_Cas** dans **Video**.
$\Pi_{Cassette}(F) = \{DF_5, DF_6, DF_7\}$

Exo 3-2

Sans perte de dépendances :
(cours) $\Pi_{R_1}(F) \cup \Pi_{R_2}(F) ... \cup \Pi_{R_k}(F) \models F$

Démonstration :(
Reprenons l'ensemble F et les relations :
(DF01-04) **No_Cli** → **Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF05-07) **No_Cas** → **Titre_Film, PrixAchat_Cas, Nom_Edi**
(DF08) **Nom_Edi** → **Ville_Edi**
(DF09-13) **Titre_Film** → **Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **Nom_Edi**, **Ville_Edi**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

$\Pi_{Video}(F) = \{DF_1 ... DF_5, DF_7 ... DF_{15}\}$ mais pas (DF6) car il manque **PrixAchat_Cas** dans **Video**.
$\Pi_{Cassette}(F) = \{DF_5, DF_6, DF_7\}$
On a donc bien $\Pi_{Video}(F) \cup \Pi_{Cassette}(F)$ qui donne tout F.
:)

Exo 3-2

Sans perte de données (disparition ou ajout)
(cours) $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$

Démonstration :(

Exo 3-2

Sans perte de données (disparition ou ajout)
(cours) $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$

Démonstration :(

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **Nom_Edi**, **Ville_Edi**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

Exo 3-2

Sans perte de données (disparition ou ajout)
(cours) $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$

Démonstration :(

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **Nom_Edi**, **Ville_Edi**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

$(Cassette \cap Video) = \{No_Cas, Titre_Film, Nom_Edi\}$
$(Cassette - Video) = \{PrixAchat_Cas\}$
$(Video - Cassette) =$
{(*No_Cli*, *Nom_Cli*, *Prenom_Cli*, *Ville_Cli*, *DateDebut_AboCli*, *Ville_Edi*, *Duree_Film*, *Nat_Film*, *Nom_Rea*, *AnnSor_Film*, *Genre*, *DateLoc_Cas*, *Duree_*

Exo 3-2

Sans perte de données (disparition ou ajout)
(cours) $(R_1 \cap R_2) \rightarrow (R_1 - R_2)$ ou $(R_1 \cap R_2) \rightarrow (R_2 - R_1)$

Démonstration :(

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **Nom_Edi**, **Ville_Edi**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

$(Cassette \cap Video) = \{No_Cas, Titre_Film, Nom_Edi\}$
$(Cassette - Video) = \{PrixAchat_Cas\}$
$(Video - Cassette) =$
$\{(No_Cli, Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli, Ville_Edi, Duree_Film, Nat_Film, Nom_Rea, AnnSor_Film, Genre, DateLoc_Cas, Duree_$

Connaissant `No_Cas`, on peut effectivement déterminer `PrixAchat_Cas` en utilisant la DF 6.
Connaissant `No_Cas` et d'autres choses, on peut toujours déterminer `PrixAchat_Cas` en utilisant la DF 6.
Conclusion : cette décomposition 2NF est sans perte de données.
:)

Exo 3-3

3. Vérifiez si les relations obtenues sont en 3NF. Si non, poursuivez la décomposition et justifiez.

3NF : 2NF et pas de DF où **"attr. non clé" → "attr. non clé"**

Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **Nom_Edi**, **Ville_Edi**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

(DF01-04) **No_Cli → Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF08) **Nom_Edi → Ville_Edi**
(DF09-13) **Titre_Film → Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre**
(DF14-15) **No_Cas, DateLoc_Cas → No_Cli, Duree_Loc**

On commence par les DF qui permettent de déterminer le plus d'attributs.

Exo 3-3

3. Vérifiez si les relations obtenues sont en 3NF. Si non, poursuivez la décomposition et justifiez.

3NF : 2NF et pas de DF où **"attr. non clé" → "attr. non clé"**

Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **Nom_Edi**, **Ville_Edi**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

(DF01-04) **No_Cli → Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli**
(DF08) **Nom_Edi → Ville_Edi**
(DF09-13) Titre_Film → Duree_Film, Nat_Film , Nom_Rea, AnnSor_Film, Genre
(DF14-15) **No_Cas, DateLoc_Cas → No_Cli, Duree_Loc**

On prend les DF (non 3 NF compatibles) 9 à 13 pour créer une relation :
Film (**Titre_Film**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**)
On constate alors que sans ces DF, les attributs concernés n'apparaissent plus dans les DF de **Video** : on peut les supprimer de cette relation.

Exo 3-3

3. Vérifiez si les relations obtenues sont en 3NF. Si non, poursuivez la décomposition et justifiez.

3NF : 2NF et pas de DF où **"attr. non clé" → "attr. non clé"**

Film (**Titre_Film**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**)
Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Nom_Edi**, **Ville_Edi**, **DateLoc_Cas**, **Duree_Loc**)

(DF01-04) No_Cli → Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli
(DF08) **Nom_Edi → Ville_Edi**
(DF14-15) **No_Cas, DateLoc_Cas → No_Cli, Duree_Loc**

On prend les DF (non 3 NF compatibles) 1 à 4 pour créer une relation :
Client (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**)
On constate que sans ces DF, les attributs concernés à part **No_Client** n'apparaissent plus dans les DF de **Video** : on peut les supprimer de cette relation.

Exo 3-3

3. Vérifiez si les relations obtenues sont en 3NF. Si non, poursuivez la décomposition et justifiez.

3NF : 2NF et pas de DF où **"attr. non clé" → "attr. non clé"**

Client (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**)
Film (**Titre_Film**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**)
Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

Video (**No_Cli**, **No_Cas**, **Nom_Edi**, **Ville_Edi**, **DateLoc_Cas**, **Duree_Loc**)

(DF08) Nom_Edi → Ville_Edi
(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

On prend la dernière DF non 3 NF compatibles : la DF08 pour créer une relation :
Editeur (**Nom_Cli**, **Ville_Edi**)
On constate que sans ces DF, les attributs concernés n'apparaissent plus dans les DF de **Video** : on peut les supprimer de cette relation.

Exo 3-3

3. Vérifiez si les relations obtenues sont en 3NF. Si non, poursuivez la décomposition et justifiez.

3NF : 2NF et pas de DF où **"attr. non clé" → "attr. non clé"**

Editeur (**Nom_Cli**, **Ville_Edi**)
Client (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**)
Film (**Titre_Film**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**)
Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)

Video (**No_Cli**, **No_Cas**, **DateLoc_Cas**, **Duree_Loc**)

(DF14-15) **No_Cas, DateLoc_Cas** → **No_Cli, Duree_Loc**

Il n'y a plus de DF problématique. Il suffit d'utiliser DF14 et DF15 pour finaliser la relation **Video** :

Video (**No_Cas, DateLoc_Cas**, **No_Cli**, **Duree_Loc**)

Exo 3-3

3. Vérifiez si les relations obtenues sont en 3NF. Si non, poursuivez la décomposition et justifiez.

3NF : 2NF et pas de DF où **"attr. non clé"** $\rightarrow$ **"attr. non clé"**

Editeur (**Nom_Cli**, **Ville_Edi**)
Client (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**)
Film (**Titre_Film**, **Duree_Film**, **Nat_Film**, **Nom_Rea**, **AnnSor_Film**, **Genre**)
Cassette (**No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**)
Video (**No_Cas**, **DateLoc_Cas**, **No_Cli**, **Duree_Loc**)

Fin de l'algorithme de normalisation.

Remarque finale

Video (**No_Cli**, **Nom_Cli**, **Prenom_Cli**, **Ville_Cli**, **DateDebut_AboCli**, **No_Cas**, **Titre_Film**, **PrixAchat_Cas**, **Nom_Edi**, Ville_Edi, **Duree_Film**, **Nat_Film** , **Nom_Rea**, **AnnSor_Film**, **Genre**, **DateLoc_Cas**, **Duree_Loc**)

1. Montrez que cette relation n'est pas en 2NF. (Identifiez les dépendances partielles : une partie de la clé détermine certains attributs.)

2NF : 1NF et pas de DF où **"bout de clé candidate"** → **"attr. non clé"**

Prouver non-2NF ? Il faut trouver la clé et une DF non 2NF-compatible.
Trouver la clé ? Il faut trouver les DF issues de cette relation.
Quelles sont les DF **logiques/évidentes** ?

```
No_Cli → Nom_Cli, Prenom_Cli, Ville_Cli, DateDebut_AboCli
No_Cas → Titre_Film, PrixAchat_Cas, Nom_Edi
Nom_Edi → Ville_Edi
Titre_Film → Duree_Film, Nat_Film, Nom_Rea, AnnSor_Film, Genre
No_Cas, DateLoc_Cas → No_Cli, Duree_Loc
No_Cli, DateLoc_Cas → No_Cas, Duree_Loc  <- Rajout ?
```

Puisque les DF n'étaient pas imposées, on peut faire les choix logiques qu'on veut, tant qu'ils sont raisonnables. En rajouter ou en supprimer, modifie bien entendu le déroulé de la normalisation.

Exercice 4

(Rappel) Méthodologie : couverture minimale

Méthodologie pour trouver une couverture minimale F_{min} :

1. Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \to X$ ou $XY \to X$).
2. Simplifier les déterminants **réductibles** ($AX \to C$ avec $A \to X$ donne $A \to C$).
3. Utiliser les règles d'Armstrong pour supprimer les DF qu'on peut retrouver en utilisant les autres DF et les règles d'Armstrong.

Astuce : surveillez les attributs recevant plusieurs arcs.

Règles d'Armstrong

R_1	Axiome de réflexivité	si	$Y \subseteq X$	alors	$X \to Y$
R_2	Axiome d'augmentation	si	$X \to Y$	alors	$XZ \to YZ$
R_3	Axiome de transitivité	si	$X \to Y$ et $Y \to Z$	alors	$X \to Z$
R_4	Règle de l'union	si	$X \to Y$ et $X \to Z$	alors	$X \to YZ$
R_5	Règle de la décomposition	si	$X \to YZ$	alors	$X \to Y$ et $X \to Z$
R_6	Règle de la composition	si	$X \to Y$ et $A \to B$	alors	$XA \to YB$
R_7	Règle de pseudo-transitivité	si	$X \to Y$ et $YZ \to W$	alors	$XZ \to W$

Exo 4

Soit la relation **R**(A, B, C, D, E).
Soit l'ensemble de DF :
F = {A → B, B → D, C → A, AD → C, CB → E, AEC → D}

Proposez une décomposition de R en 3NF, garantissant :

- aucune perte de dépendances fonctionnelles
- aucune perte de données

Indications :

1. Calcul de la couverture minimale,
2. Détermination des clés,
3. Application de l'algorithme de décomposition en 3NF

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) $A \rightarrow B$

(DF2) $B \rightarrow D$

(DF3) $C \rightarrow A$

(DF4) $AD \rightarrow C$

(DF5) $CB \rightarrow E$

(DF6) $AEC \rightarrow D$

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) A $\rightarrow$ B
(DF2) B $\rightarrow$ D
(DF3) C $\rightarrow$ A
(DF4) AD $\rightarrow$ C
(DF5) CB $\rightarrow$ E
(DF6) AEC $\rightarrow$ D

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) $A \rightarrow B$
(DF2) $B \rightarrow D$
(DF3) $C \rightarrow A$
(DF4) $AD \rightarrow C$
(DF5) $CB \rightarrow E$
(DF6) $AEC \rightarrow D$

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).

Rien à faire : tout est déjà en ordre.

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) $A \rightarrow B$

(DF2) $B \rightarrow D$

(DF3) $C \rightarrow A$

(DF4) $AD \rightarrow C$

(DF5) $CB \rightarrow E$

(DF6) $AEC \rightarrow D$

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).

  Rien à faire : tout est déjà en ordre.

- Simplifier les déterminants **réductibles** ($AX \rightarrow C$ avec $A \rightarrow X$ donne $A \rightarrow C$).

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) A → B
(DF2) B → D
(DF3) C → A
(DF4) AD → C
(DF5) CB → E
(DF6) AEC → D

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).

  Rien à faire : tout est déjà en ordre.

- Simplifier les déterminants **réductibles** ($AX \rightarrow C$ avec $A \rightarrow X$ donne $A \rightarrow C$).
  (DF4) $A^+ = \{ A, B, D \}$. Donc A suffisant dans AD : $A \rightarrow C$.

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) A → B

(DF2) B → D

(DF3) C → A

(DF4) AD → C devient A → C

(DF5) CB → E

(DF6) AEC → D

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).

  Rien à faire : tout est déjà en ordre.

- Simplifier les déterminants **réductibles** ($AX \rightarrow C$ avec $A \rightarrow X$ donne $A \rightarrow C$).

  (DF4) $A^+ = \{ A, B, D \}$. Donc A suffisant dans AD : $A \rightarrow C$.

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) A → B
(DF2) B → D
(DF3) C → A
(DF4) AD → C devient A → C
(DF5) CB → E
(DF6) AEC → D

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).
  Rien à faire : tout est déjà en ordre.
- Simplifier les déterminants **réductibles** ($AX \rightarrow C$ avec $A \rightarrow X$ donne $A \rightarrow C$).
  (DF4) $A^+ = \{ A, B, D \}$. Donc A suffisant dans AD : $A \rightarrow C$.
  (DF5) $C^+ = \{ C, A, B, D \}$. Donc C suffisant dans CB : $C \rightarrow E$.

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) A → B
(DF2) B → D
(DF3) C → A
(DF4) AD → C devient A → C
(DF5) CB → E devient C → E
(DF6) AEC → D

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \rightarrow X$ ou $XY \rightarrow X$).

  Rien à faire : tout est déjà en ordre.

- Simplifier les déterminants **réductibles** ($AX \rightarrow C$ avec $A \rightarrow X$ donne $A \rightarrow C$).
  (DF4) $A^+ = \{ A, B, D \}$. Donc A suffisant dans AD : $A \rightarrow C$.
  (DF5) $C^+ = \{ C, A, B, D \}$. Donc C suffisant dans CB : $C \rightarrow E$.

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) A → B
(DF2) B → D
(DF3) C → A
(DF4) AD → C devient A → C
(DF5) CB → E devient C → E
(DF6) AEC → D

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \to X$ ou $XY \to X$).
  Rien à faire : tout est déjà en ordre.
- Simplifier les déterminants **réductibles** ($AX \to C$ avec $A \to X$ donne $A \to C$).
  (DF4) $A^+ = \{ A, B, D \}$. Donc A suffisant dans AD : $A \to C$.
  (DF5) $C^+ = \{ C, A, B, D \}$. Donc C suffisant dans CB : $C \to E$.
  (DF6) $C^+ = \{ C, A, B, D, E \}$. Donc C suffisant dans AEC : $C \to D$ ou
  $A^+ = \{ A, C, B, D, E \}$. Donc A suffisant dans AEC : $A \to D$ au choix !

Exo 4

R(A, B, C, D, E) munie de l'ensemble F de DF suivant :

(DF1) A → B

(DF2) B → D

(DF3) C → A

(DF4) AD → C devient A → C

(DF5) CB → E devient C → E

(DF6) AEC → D devient A → D par choix

On cherche d'abord à trouver la **couverture minimale de DF** permettant de conserver les mêmes propriétés mais avec un ensemble de DF plus réduite et/ou simplifié.

1/3 Trouver une couverture minimale F_{min} :

- Transformer les DP à l'aide de la règle de décomposition pour n'avoir que des déterminés atomiques et supprimer les DP triviales ($X \to X$ ou $XY \to X$).

  Rien à faire : tout est déjà en ordre.

- Simplifier les déterminants **réductibles** ($AX \to C$ avec $A \to X$ donne $A \to C$).

  (DF4) $A^+ = \{ A, B, D \}$. Donc A suffisant dans AD : $A \to C$.

  (DF5) $C^+ = \{ C, A, B, D \}$. Donc C suffisant dans CB : $C \to E$.

  (DF6) $C^+ = \{ C, A, B, D, E \}$. Donc C suffisant dans AEC : $C \to D$ ou $A^+ = \{ A, C, B, D, E \}$. Donc A suffisant dans AEC : $A \to D$ au choix !

Exo 4

R(A, B, C, D, E) munie de l'ensemble F réduit suivant :

(DF1) A → B
(DF2) B → D
(DF3) C → A
(DF4) A → C
(DF5) C → E
(DF6) A → D

Notez que d'autres choix peuvent donner d'autres F réduites.

1/3 Trouver une couverture minimale F_{min} :

- Utiliser les règles d'Armstrong pour supprimer les DF qu'on peut retrouver en utilisant les autres DF et les règles d'Armstrong.

Astuce : surveillez les attributs recevant plusieurs arcs.

Règles d'Armstrong

R_1	Axiome de réflexivité	si	$Y \subseteq X$	alors	$X \to Y$
R_2	Axiome d'augmentation	si	$X \to Y$	alors	$XZ \to YZ$
R_3	Axiome de transitivité	si	$X \to Y$ et $Y \to Z$	alors	$X \to Z$
R_4	Règle de l'union	si	$X \to Y$ et $X \to Z$	alors	$X \to YZ$
R_5	Règle de la décomposition	si	$X \to YZ$	alors	$X \to Y$ et $X \to Z$
R_6	Règle de la composition	si	$X \to Y$ et $A \to B$	alors	$XA \to YB$
R_7	Règle de pseudo-transitivité	si	$X \to Y$ et $YZ \to W$	alors	$XZ \to W$

Exo 4

R(A, B, C, D, E) munie de l'ensemble F réduit suivant :

(DF1)	A → B	
(DF2)	B → D	détermine D
(DF3)	C → A	
(DF4)	A → C	
(DF5)	C → E	
(DF6)	A → D	détermine D

Notez que d'autres choix peuvent donner d'autres F réduites.

1/3 Trouver une couverture minimale F_{min} :

- Utiliser les règles d'Armstrong pour supprimer les DF qu'on peut retrouver en utilisant les autres DF et les règles d'Armstrong.

Astuce : surveillez les attributs recevant plusieurs arcs.

Règles d'Armstrong

R_1	Axiome de réflexivité	si	$Y \subseteq X$	alors	$X \to Y$
R_2	Axiome d'augmentation	si	$X \to Y$	alors	$XZ \to YZ$
R_3	Axiome de transitivité	si	$X \to Y$ et $Y \to Z$	alors	$X \to Z$
R_4	Règle de l'union	si	$X \to Y$ et $X \to Z$	alors	$X \to YZ$
R_5	Règle de la décomposition	si	$X \to YZ$	alors	$X \to Y$ et $X \to Z$
R_6	Règle de la composition	si	$X \to Y$ et $A \to B$	alors	$XA \to YB$
R_7	Règle de pseudo-transitivité	si	$X \to Y$ et $YZ \to W$	alors	$XZ \to W$

Exo 4

R(A, B, C, D, E) munie de l'ensemble F réduit suivant :

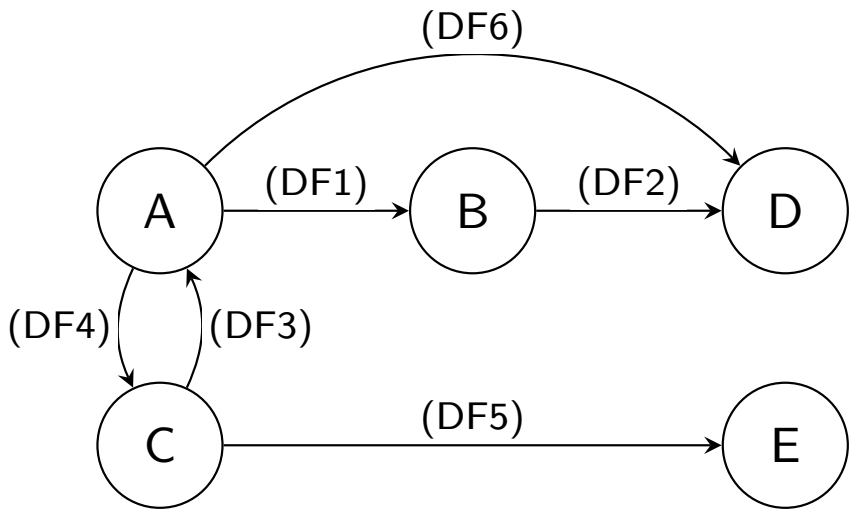
(DF1) A → B
(DF2) B → D
(DF3) C → A
(DF4) A → C
(DF5) C → E
(DF6) A → D

Notez que d'autres choix peuvent donner d'autres F réduites.

On cherche ici à éliminer les DF non nécessaires car retrouvables à partir des autres.

Comme le graphe a l'air simple, vous pouvez le tracer sur votre brouillon.

On visualise immédiatement que DF6 n'est pas nécessaire : on peut la supprimer sans perdre de dépendances puisqu'on peut utiliser la transitivité pour retrouver $A \rightarrow D$.



Exo 4

R(A, B, C, D, E) munie de l'ensemble F réduit suivant :

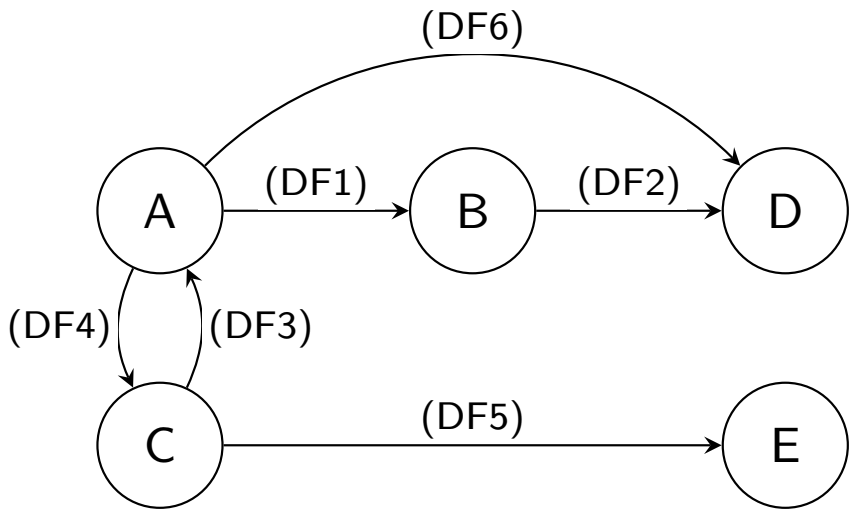
(DF1)	A → B	
(DF2)	B → D	détermine D
(DF3)	C → A	
(DF4)	A → C	
(DF5)	C → E	
(DF6)	A → D	détermine D

Notez que d'autres choix peuvent donner d'autres F réduites.

On cherche ici à éliminer les DF non nécessaires car retrouvables à partir des autres.

Comme le graphe a l'air simple, vous pouvez le tracer sur votre brouillon.

On visualise immédiatement que DF6 n'est pas nécessaire : on peut la supprimer sans perdre de dépendances puisqu'on peut utiliser la transitivité pour retrouver $A \rightarrow D$.



Exo 4

R(A, B, C, D, E) munie de l'ensemble F réduit suivant :

(DF1) $A \rightarrow B$
(DF2) $B \rightarrow D$
(DF3) $C \rightarrow A$
(DF4) $A \rightarrow C$
(DF5) $C \rightarrow E$
(DF6) $A \rightarrow D$

Notez que d'autres choix peuvent donner d'autres F réduites.

On cherche ici à éliminer les DF non nécessaires car retrouvables à partir des autres.

Exo 4

R(A, B, C, D, E) munie de l'ensemble F réduit suivant :

(DF1)	A → B	
(DF2)	B → D	détermine D
(DF3)	C → A	
(DF4)	A → C	
(DF5)	C → E	
(DF6)	A → D	détermine D

Notez que d'autres choix peuvent donner d'autres F réduites.

On cherche ici à éliminer les DF non nécessaires car retrouvables à partir des autres.

Montrons que (DF1+DF2) donne (DF6)

(DF6 n'est donc pas nécessaire dans la couverture minimale)

Exo 4

R(A, B, C, D, E) munie de l'ensemble F réduit suivant :

(DF1)	A → B	
(DF2)	B → D	détermine D
(DF3)	C → A	
(DF4)	A → C	
(DF5)	C → E	
(DF6)	A → D	détermine D

Notez que d'autres choix peuvent donner d'autres F réduites.

On cherche ici à éliminer les DF non nécessaires car retrouvables à partir des autres.

Montrons que (DF1+DF2) donne (DF6)

(DF6 n'est donc pas nécessaire dans la couverture minimale)

:(

$$(A \rightarrow B) \ \wedge \ (B \rightarrow D) \underset{\text{transitivité}}{\Longrightarrow} (A \rightarrow D)$$

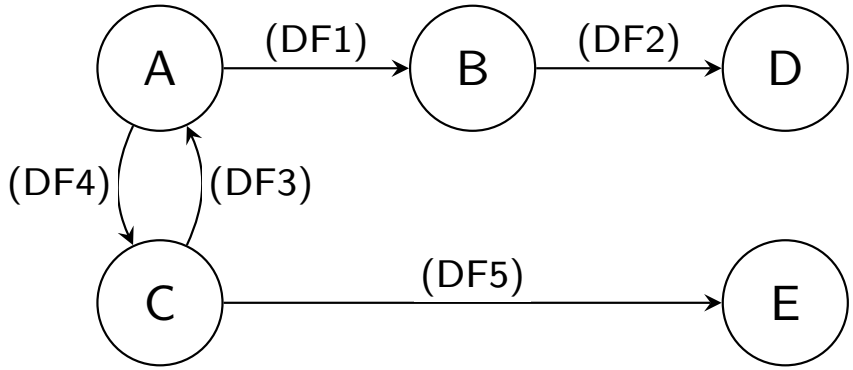
Conclusion : la DF6 n'est pas nécessaire à notre ensemble de DF minimale.

:)

Exo 4

On obtient la couverture minimale F_{min} suivante :

(DF1) A → B
(DF2) B → D
(DF3) C → A
(DF4) A → C
(DF5) C → E



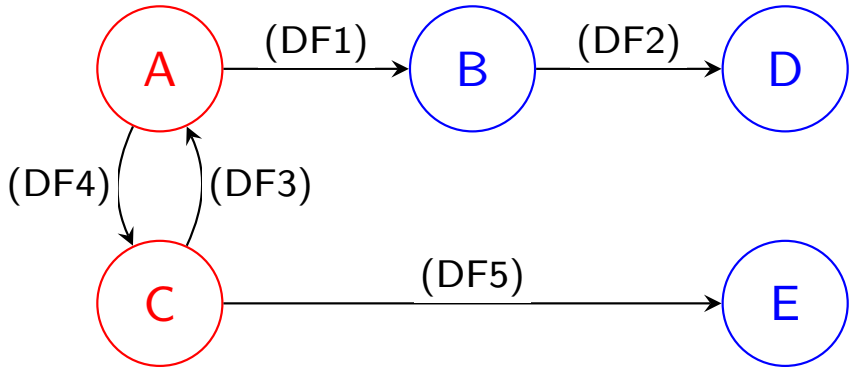
2/3 Recherche des clés candidates

Aucun n'attribut n'apparaît jamais à droite...
Il faut donc chercher parmi ceux qui apparaissent à gauche.
$A^+ = \{ A, \quad C, B, \quad D, E \} = ABCDE$
$B^+ = \{ B, \quad D \} = BD$
$C^+ = \{ C, \quad A, D, \quad B, \quad E \} = ABCDE$
Ces deux clés candidates A et C sont minimales puisqu'elles ne comportent qu'un attribut.
Cette couverture minimale possède donc deux clés candidates : A et C.

Exo 4

On considère donc **R(A, B, C, D, E)** :

(DF1) **A** → **B**
(DF2) **B** → **D**
(DF3) **C** → **A**
(DF4) **A** → **C**
(DF5) **C** → **E**



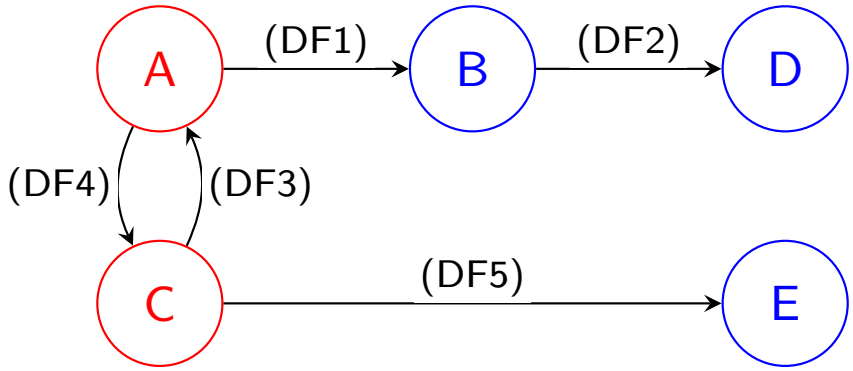
3/3 Algo de décomposition

On commence par utiliser les attributs A ou C car ce sont eux qui déterminent le plus d'autres attributs.

Exo 4

On considère donc **R(A, B, C, D, E)** :

(DF1) **A** → **B**
(DF2) **B** → **D**
(DF3) **C** → **A**
(DF4) **A** → **C**
(DF5) **C** → **E**

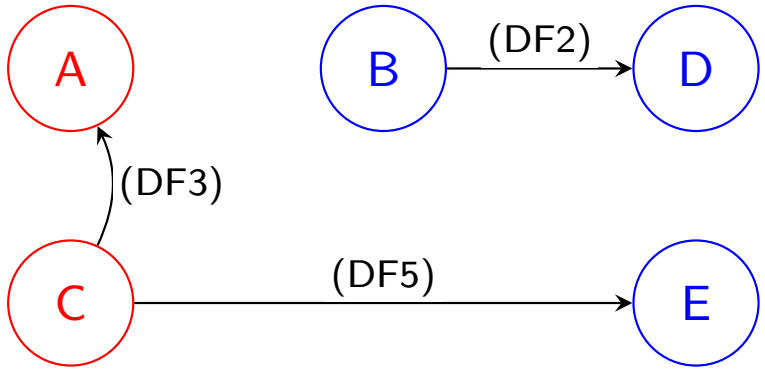


3/3 Algo de décomposition

On commence par utiliser les attributs A ou C car ce sont eux qui déterminent le plus d'autres attributs.

- **R1(A, C)** à partir de DF4.
- **R2[A, B)** à partir de DF1.

Aucun sommet ne devient isolé.
On a toujours
R(A, B, C, D, E)

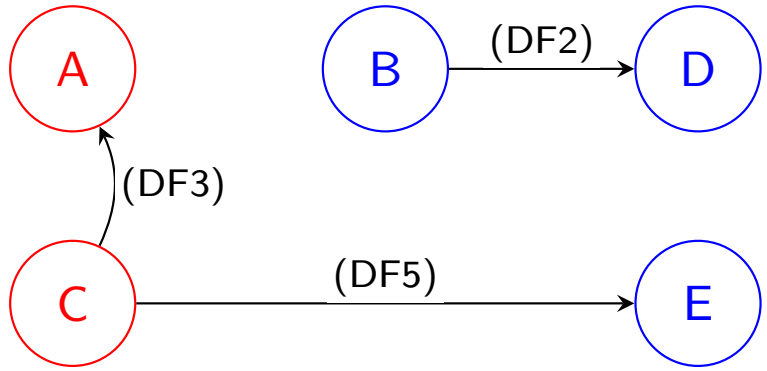


Exo 4

On considère maintenant **R(A, B, C, D, E)** :

(DF2) **B** → **D**
(DF3) **C** → **A**

(DF5) **C** → **E**



3/3 Algo de décomposition

- **R1(A, C)** à partir de DF4.
- **R2[A, B)** à partir de DF1.

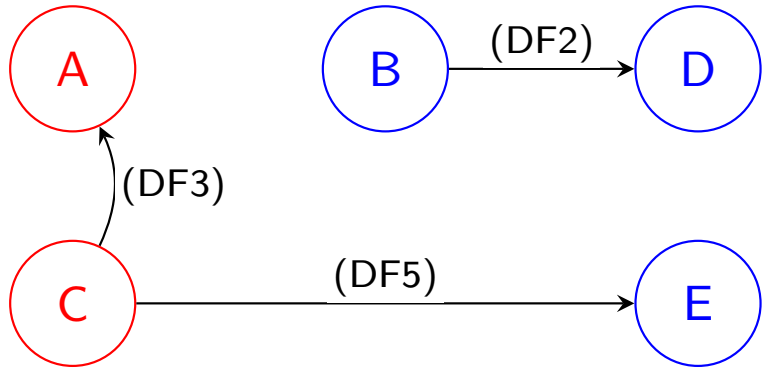
On choisit C puisqu'il détermine deux attributs.

Exo 4

On considère maintenant **R(A, B, C, D, E)** :

(DF2) **B → D**
(DF3) **C → A**

(DF5) **C → E**



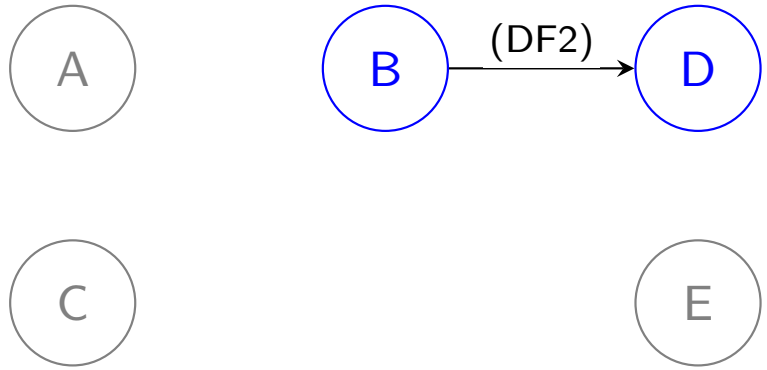
3/3 Algo de décomposition

- **R1(A, C)** à partir de DF4.
- **R2[A, B)** à partir de DF1.

On choisit C puisqu'il détermine deux attributs.

- **R3(C, A)** à partir de DF3.
- **R4[C, E)** à partir de DF5.

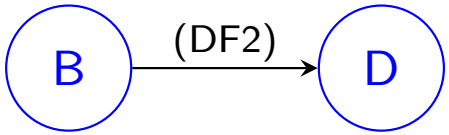
C, E et A deviennent isolés.
R(B, D)



Exo 4

On considère maintenant **R(B, D)** :

(DF2) **B** $\rightarrow$ **D**



3/3 Algo de décomposition

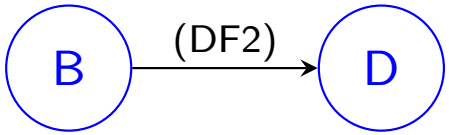
- **R1**(**A**, **C**) à partir de DF4.
- **R2**[**A**, **B**) à partir de DF1.
- **R3**(**C**, **A**) à partir de DF3.
- **R4**[**C**, **E**) à partir de DF5.

Il ne reste que DF 2.

Exo 4

On considère maintenant **R(B, D)** :

(DF2) $\quad B \rightarrow D$



3/3 Algo de décomposition

- **R1(A, C)** à partir de DF4.
- **R2[A, B)** à partir de DF1.
- **R3(C, A)** à partir de DF3.
- **R4[C, E)** à partir de DF5.



Il ne reste que DF 2.

- **R5[B, D)** à partir de DF2.

B, D deviennent isolés.

R(∅)

Exo 4

Fin ?

Il reste maintenant à savoir si il faut regrouper ou pas les relations qui ont les mêmes clés primaires puis les nommer. Mais il faut avoir le contexte (que nous n'avons pas ici).

Possibilité 1 :

- **R1**(**A**, **C**) à partir de DF4.
- **R2**[**A**, **B**) à partir de DF1.
- **R3**(**C**, **A**) à partir de DF3.
- **R4**[**C**, **E**) à partir de DF5.
- **R5**[**B**, **D**) à partir de DF2.

Reste le problème de la boucle A-C qui peut poser des problèmes de cohérence lors des mises à jour.
Plutôt que de laisser ce problème, on peut choisir de supprimer la boucle et de choisir qui de A ou C détermine l'autre.

Possibilité 2 :

- **R12**(**A**, **C**, **B**)
- **R34**(**C**, **A**, **E**)
- **R5**[**B**, **D**)

Possibilité 3 :

- **R1**(**A**, **C**) à partir de DF4.
- **R2**[**A**, **B**) à partir de DF1.
- suppression de R3
- **R4**[**C**, **E**) à partir de DF5.
- **R5**[**B**, **D**) à partir de DF2.