

BCNF

a) Appliquer les formes normales

Étape 1 Première forme normale : La table actuelle respecte déjà ces critères, car chaque cellule contient une seule valeur, et il n'y a pas de colonnes multivaluées. Ainsi, elle est déjà en 1NF. Aucune transformation n'est nécessaire pour cette étape.

Étape 2 deuxième forme normale : Pour être en 2NF, la table doit être en 1NF et tous les attributs non-clés doivent dépendre entièrement de la clé primaire.

On énumère les attributs

- NO_CHAUFFEUR
- NOM_CHAU
- NO_PARCOURS
- TYPE_PAR, DATE
- HEURE_DEBUT_PAR
- HEURE_FIN_PAR
- NO_AUTOBUS

La table de départ est donc:

HORAIRES_CHAUFFEURS_AUTOBUS(NO_CHAUFFEUR,NOM_CHAU,
NO_PARCOURS, TYPE_PAR, DATE, HEURE_DEBUT_PAR, HEURE_FIN_PAR,
NO_AUTOBUS)

Dépendances fonctionnelles (DF) identifiées :

1. Df1 : NO_CHAUFFEUR → NOM_CHAU
Un chauffeur est identifié par un numéro unique et a un nom.
2. Df2: NO_PARCOURS → TYPE_PAR
Un numéro de parcours correspond à un type de parcours.
3. Df3 : NO_PARCOURS, DATE_PAR, HEURE_DEBUT_PAR → NOM_CHAU,
NO_CHAUFFEUR, TYPE_PAR, HEURE_FIN_PAR, NO_AUTOBUS
Un parcours, pour une date et une heure de début, est associé à un bus, un chauffeur et d'autres détails, on pourrait aussi permuter entre HEURE_FIN_PAR et HEURE_DEBUT_PAR pour en faire une autre df
4. Df4 : NO_AUTOBUS, DATE_PAR, HEURE_DEBUT_PAR → NO_CHAUFFEUR,
NOM_CHAU, NO_PARCOURS, TYPE_PAR, HEURE_FIN_PAR
Pour un bus donné, à une date et heure de début, on peut retrouver toutes les autres informations, on pourrait aussi permuter entre HEURE_FIN_PAR et HEURE_DEBUT_PAR pour en faire une autre df
5. Df5 : NO_CHAUFFEUR, DATE_PAR, HEURE_DEBUT_PAR → NO_AUTOBUS,
NOM_CHAU, NO_PARCOURS, TYPE_PAR, HEURE_FIN_PAR

Similairement, l'heure de fin pour un bus donné permet de retrouver toutes les autres informations, on pourrait aussi permuter entre HEURE_FIN_PAR et HEURE_DEBUT_PAR pour en faire une autre df

On identifie les clés candidates qui nous permettent de retrouver tous les éléments d'identification **des clés candidates** :

Les clés candidates sont les ensembles minimaux d'attributs qui permettent de déterminer tous les attributs de la table. D'après les DF ci-dessus, voici les déterminants potentiels :

- **DF3** : NO_PARCOURS, DATE_PAR, HEURE_DEBUT_PAR
- **DF4** : NO_AUTOBUS, DATE_PAR, HEURE_DEBUT_PAR
- **DF5** : NO_CHAUFFEUR, DATE_PAR, HEURE_DEBUT_PAR

Clés candidates :

- NO_PARCOURS, DATE_PAR, HEURE_DEBUT_PAR (choisie comme clé primaire)
- NO_AUTOBUS, DATE, HEURE_DEBUT_PAR (clé alternative)
- NO_CHAUFFEUR, DATE, HEURE_DEBUT_PAR (clé alternative)

Table normalisée (2NF) :

HORAIRES_CHAUFFEURS_AUTOBUS(NO_CHAUFFEUR, DATE, HEURE_DEBUT_PAR, NOM_CHAU, NO_PARCOURS, TYPE_PAR, HEURE_FIN_PAR, NO_AUTOBUS)

Élimination des dépendances partielles :

La DF1 (NO_CHAUFFEUR → NOM_CHAU) est une dépendance partielle, car NOM_CHAU dépend uniquement de NO_CHAUFFEUR, une partie de la clé primaire (NO_CHAUFFEUR, DATE, HEURE_DEBUT_PAR). Cela viole la 2NF.

Pour résoudre cela, il faut déplacer NOM_CHAU dans une nouvelle table, avec NO_CHAUFFEUR comme clé primaire. Cette nouvelle table sera référencée dans la table principale via une clé étrangère.

Nouvelles tables après passage en 2NF :

1. **CHAUFFEUR (NO_CHAUFFEUR, NOM_CHAU)**

NO_CHAUFFEUR	NOM_CHAU
C40454	Louis Tremblay
C40489	Nicole Simard
C41112	Simon Gauvin

Cette table contient des informations spécifiques aux chauffeurs.

2. **HORAIRES_CHAUFFEURS_AUTOBUS** (NO_CHAUFFEUR#, DATE, HEURE_DEBUT_PAR, NO_PARCOURS, TYPE_PAR, HEURE_FIN_PAR, NO_AUTOBUS)

NO_CHAUFFEUR	NO_PARCOURS	TYPE_PAR	DATE_PAR	HEURE_DEBUT_PAR	HEURE_FIN_PAR	NO_AUTOBUS
C40454	80	Régulier	2015-10-15	8 :00	12 :00	LFS43308
C40489	7	Régulier	2015-10-15	12 :00	18 :00	LFS44410
C40454	280	Express	2015-10-15	12 :15	16 :00	LFS43308
C40489	11	Régulier	2015-10-16	12 :00	18 :00	LFS44411
C41112	338	Rapido	2015-10-16	8 :00	16 :00	LGG44412
C41112	338	Rapido	2015-10-17	9 :00	17 :00	LGG44413
C41112	7	Régulier	2015-10-18	9 :00	18 :00	LGG44412
C40489	7	Régulier	2015-10-18	16 :00	22 :00	LGH33322

Cette table contient les informations sur les horaires des chauffeurs, leurs bus, et leurs parcours.

Étape 3 : Passage de la 2NF à la 3NF

La troisième forme normale (3NF) exige que, pour chaque dépendance fonctionnelle (DF) dans une table, tout attribut non-clé doit dépendre directement de la clé primaire. Il ne doit donc pas y avoir de dépendances transitives, c'est-à-dire qu'un attribut non-clé ne doit pas dépendre d'un autre attribut non-clé.

Identification des dépendances fonctionnelles (DF) pour chaque table

Table CHAUFFEUR

- DF1 : NO_CHAUFFEUR → NOM_CHAU
Un chauffeur est identifié par un numéro unique, et cela détermine son nom.

Cette table ne contient qu'une seule dépendance fonctionnelle, ce qui signifie qu'il n'y a pas de dépendance transitive possible.

Table HORAIRES_CHAUFFEURS_AUTOBUS

- DF2: NO_PARCOURS → TYPE_PAR
Un numéro de parcours détermine un type de parcours.
- DF3 : NO_PARCOURS, DATE_PAR, HEURE_DEBUT_PAR → NO_CHAUFFEUR, NOM_CHAU, TYPE_PAR, HEURE_FIN_PAR,

NO_AUTOBUS

Le numéro du parcours, pour une date et une heure de début, est associé à un bus, un parcours et d'autres détails.

- DF4 : NO_AUTOBUS, DATE_PAR, HEURE_DEBUT_PAR → NO_CHAUFFEUR, NOM_CHAU, NO_PARCOURS, TYPE_PAR, HEURE_FIN_PAR
Pour un bus donné, à une date et une heure de début, on peut retrouver toutes les autres informations.
- DF5: NO_CHAUFFEUR, DATE_PAR, HEURE_DEBUT_PAR → NO_AUTOBUS, NOM_CHAU, NO_PARCOURS, TYPE_PAR, HEURE_FIN_PAR
Similairement, l'heure de fin pour un bus donné permet de retrouver toutes les autres informations.
- **DF6** : NO_CHAUFFEUR, DATE_PAR → NO_AUTOBUS

Identification des clés primaires (PK) pour chaque table

- **CHAUFFEUR:**
La clé primaire (PK) est **NO_CHAUFFEUR**.
- **HORAIRES_CHAUFFEURS_AUTOBUS** :
La clé primaire (PK) est **NO_PARCOURS, DATE, HEURE_DEBUT_PAR**.

Identification des dépendances transitives

Pour passer en 3NF, il faut éliminer toute dépendance transitive, c'est-à-dire lorsqu'un attribut non-clé dépend d'un autre attribut non-clé à travers une relation indirecte.

Table CHAUFFEUR

Dans la table **CHAUFFEUR**, il n'y a qu'une seule dépendance fonctionnelle :

- **DF1** : NO_CHAUFFEUR → NOM_CHAU

Aucune dépendance transitive n'est présente dans cette table, donc elle est déjà en 3NF.

Table HORAIRES_CHAUFFEURS_AUTOBUS

Dans cette table, il y a une dépendance transitive à identifier.

- **Df2** : NO_PARCOURS → TYPE_PAR
- DF3 peut être décomposée en deux dépendances :
 - **Df3.1** : NO_PARCOURS, DATE_PAR, HEURE_DEBUT_PAR → TYPE_PAR
Cela signifie que **NO_PARCOURS** dépend directement de la clé primaire.

- **Df3.2** : NO_PARCOURS → TYPE_PAR
Cela signifie que **TYPE_PAR** dépend de **NO_PARCOURS**, ce qui crée une dépendance transitive avec la clé primaire

Transformation en 3NF

Pour éliminer cette dépendance transitive dans la table **HORAIRE_CHAUFFEURS_AUTOBUS**, nous devons déplacer la dépendance **NO_PARCOURS → TYPE_PAR** dans une nouvelle table. Cela nous donne les relations suivantes :

1. Table CHAUFFEUR (NO_CHAUFFEUR, NOM_CHAU)

NO_CHAUFFEUR	NOM_CHAU
C40454	Louis Tremblay
C40489	Nicole Simard
C41112	Simon Gauvin

Cette table reste inchangée.

2. Table PARCOURS (NO_PARCOURS, TYPE_PAR)

NO_PARCOURS	TYPE_PAR
80	Régulier
7	Régulier
280	Express
11	Régulier
338	Rapido

Cette nouvelle table contient les informations sur les types de parcours, ce qui élimine la dépendance transitive dans **HORAIRE_CHAUFFEURS_AUTOBUS**.

3. HORAIRE_PARCOURS (NO_PARCOURS#, DATE_PAR, HEURE_DEBUT_PAR, HEURE_FIN_PAR, NO_CHAUFFEUR#, NO_AUTOBUS)

NO_PARCOURS	DATE_PAR	HEURE_DEBUT_PAR	HEURE_FIN_PAR	NO_CHAUFFEUR	NO_AUTOBUS
80	2015-10-15	8 :00	12 :00	C40454	LFS43308
7	2015-10-15	12 :00	18 :00	C40489	LFS44410
280	2015-10-15	12 :15	16 :00	C40454	LFS43308
11	2015-10-16	12 :00	18 :00	C40489	LFS44411
338	2015-10-16	8 :00	16 :00	C41112	LGG44412

338	2015-10-17	9 :00	17 :00	C41112	LGG44413
7	2015-10-18	9 :00	18 :00	C41112	LGG44412
7	2015-10-18	16 :00	22 :00	C40489	LGH33322

Identification des clés primaire et les clés alternatives

1. Table CHAUFFEUR (NO_CHAUFFEUR, NOM_CHAU)

- **Clé primaire (PK)** : NO_CHAUFFEUR
- **Clé alternative (AK)** : Aucune, car le numéro du chauffeur identifie de manière unique chaque enregistrement.

2. Table PARCOURS (NO_PARCOURS, TYPE_PAR)

- **Clé primaire (PK)** : NO_PARCOURS
- **Clé alternative (AK)** : Aucune, car le numéro de parcours est unique et identifie chaque type de parcours.

3. Table HORAIRE_PARCOURS (NO_PARCOURS#, DATE_PAR, HEURE_DEBUT_PAR, HEURE_FIN_PAR, NO_CHAUFFEUR#, NO_AUTOBUS)

- **Clé primaire (PK)** : (NO_CHAUFFEUR#, DATE_PAR, HEURE_DEBUT_PAR)
- **Clés alternatives (AK)** :
 1. (NO_CHAUFFEUR, DATE_PAR, HEURE_FIN_PAR)
 2. (NO_AUTOBUS, DATE_PAR, HEURE_DEBUT_PAR)

b) Vérification de la conformité à la BCNF

Pour qu'une table respecte la BCNF (Boyce-Codd Normal Form), tous les déterminants de ses dépendances fonctionnelles (DF) doivent être des clés candidates.

i) Identification de la table ne respectant pas la BCNF

Parmi les trois tables, HORAIRE_PARCOURS est celle qui possède plusieurs dépendances fonctionnelles (DF) et des clés alternatives, ce qui la rend potentiellement non conforme à la BCNF. Les autres tables, CHAUFFEUR et PARCOURS, ne posent pas de problème car elles ne possèdent qu'une seule DF, et leurs clés primaires sont les seuls déterminants.

ii) Vérification des déterminants dans HORAIRE_PARCOURS

1. Énumération des DF (Dépendances Fonctionnelles) dans la table HORAIRE_PARCOURS :

- **DF1** : NO_PARCOURS, DATE_PAR, HEURE_DEBUT_PAR est une clé alternative donc est une clé candidat
- **DF2** : NO_AUTOBUS, DATE_PAR, HEURE_DEBUT_PAR est une clé alternative donc est une clé candidat
- **DF3** : NO_CHAUFFEUR, DATE_PAR, HEURE_DEBUT_PAR est une clé alternative donc est une clé candidat
- **DF4** : NO_CHAUFFEUR, DATE_PAR n'est pas une clé alternative donc elle n'est pas une clé candidat

2. Énumération des déterminants :

- **DF1** : NO_PARCOURS, DATE_PAR, HEURE_DEBUT_PAR
- **DF2** : NO_AUTOBUS, DATE_PAR, HEURE_DEBUT_PAR
- **DF3** : NO_CHAUFFEUR, DATE_PAR, HEURE_DEBUT_PAR
- **DF4** : NO_CHAUFFEUR, DATE_PAR

3. Identification d'une DF dont le déterminant n'est pas une clé candidate :

- Les DF1, DF2, et DF3, ont pour déterminants des clés alternatives (CK), ce qui respecte la BCNF.
- **DF4 (NO_CHAUFFEUR, DATE_PAR → NO_AUTOBUS)** est problématique : **NO_CHAUFFEUR, DATE_PAR** n'est pas une clé candidate dans la table HORAIRE_PARCOURS. Cette DF ne respecte pas la BCNF

Afin de respecter la BCNF, il faut retirer la dépendance fonctionnelle problématique DF4 de la table **HORAIRE_PARCOURS** et la placer dans une nouvelle table.

Cela donnera la nouvelle table suivante : CHAUFFEUR_AUTOBUS(**NO_CHAUFFEUR, DATE_PAR, NO_AUTOBUS**)

NO_CHAUFFEUR	DATE_PAR	NO_AUTOBUS
C40454	2015-10-15	LFS43308
C40489	2015-10-15	LFS44410
C40489	2015-10-16	LFS44411
C41112	2015-10-16	LGG44412
C41112	2015-10-17	LGG44413
C41112	2015-10-18	LGG44412
C40489	2015-10-18	LGH33322

La clé étrangère de cette table est NO_CHAUFFEUR

2) 4NF

a) Expliquez pourquoi le tableau 1 est en BCNF, mais pas en 4NF

Dans le tableau 1, nous constatons que **NO_EMPLOYE** et **NO_CLIENT** sont tous deux multivalués par rapport à **DEPARTEMENT**. Un employé peut être lié à plusieurs clients, et un client peut être lié à plusieurs employés au sein du même département. Ces dépendances multivaluées, en l'absence d'une dépendance directe entre **NO_EMPLOYE** et **NO_CLIENT**, violent la 4NF. Ainsi, le tableau 1 est en BCNF, mais pas en 4NF.

b) Processus de normalisation en 4NF du Tableau 1

Pour normaliser le tableau 1 en 4NF, nous devons supprimer les dépendances multivaluées en décomposant la table en deux relations distinctes, chacune capturant une seule de ces dépendances multivaluées.

1. Étape 1 : Identifier les dépendances multivaluées

- **DEPARTEMENT** →→ **NO_EMPLOYE**
- **DEPARTEMENT** →→ **NO_CLIENT**

Ces deux dépendances montrent que, pour chaque département, il y a une liste d'employés et une liste de clients, mais les employés et les clients ne sont pas directement liés entre eux.

2. Étape 2 : Créer des tables séparées pour chaque dépendance multivaluée

Table 1 : Relation entre DEPARTEMENT et NO_EMPLOYE

DEPARTEMENT	NO_EMPLOYE
Quincaillerie	E101
Quincaillerie	E104

Table 2 : Relation entre DEPARTEMENT et NO_CLIENT

DEPARTEMENT	NO_CLIENT
Quincaillerie	43
Quincaillerie	654

3. Étape 3 : Résultat après décomposition en 4NF

Les deux nouvelles tables éliminent les dépendances multivaluées tout en maintenant les relations entre les employés, les clients et les départements. Elles sont désormais en 4NF car il n'y a plus de dépendances multivaluées non triviales.

3) Algèbre relationnelle

a) Filtrer et afficher les titres et le nombre en inventaire des livres dont le prix de vente est compris entre 10 et 30

- Requête algébrique :

$\pi_{TITR_LIV, NB_INVENTAIRE_LIV} (\sigma_{MNT_PRIX_VENT \geq 10 \text{ et } MNT_PRIX_VENTE_LIV \leq 30} (LIVRE))$

- Explication :
 - La sélection σ filtre les lignes de la table LIVRE où le prix de vente est compris entre 10 et 30.
 - La projection π affiche uniquement les colonnes TITRE_LIV et NB_INVENTAIRE_LIV des livres répondant à ce critère.
-

b) Opérateurs d'union et d'intersection

i) Listez l'ensemble des téléphones (client ou maison d'édition) avec l'opérateur d'union \cup

- Requête algébrique :

$\pi_{TEL_CLI}(\text{CLIENT}) \cup \pi_{TEL_MAI}(\text{MAISON_EDITION})$

- Explication :
 - Nous utilisons l'opérateur d'union \cup pour combiner les téléphones des clients et des maisons d'édition.
 - Les projections π sont utilisées pour extraire les colonnes TEL_CLI et TEL_MAI de chaque table respective.

ii) Listez les courriels communs entre maisons d'édition et clients avec l'opérateur d'intersection \cap

- Requête algébrique :

$\pi_{COURRIEL_CLI}(\text{CLIENT}) \cap \pi_{COURRIEL_MAI}(\text{MAISON_EDITION})$

- Explication :
 - Nous utilisons l'opérateur d'intersection \cap pour trouver les courriels communs entre les clients et les maisons d'édition.
 - Les projections π sont utilisées pour extraire les colonnes COURRIEL_CLI et COURRIEL_MAI des deux tables.

c) Jointures

i) Listez les noms de toutes les maisons d'édition et, si disponible, la date d'achat d'un livre

- Requête algébrique :

$\pi_{NOM_MAI, DATE_ACH}(\text{MAISON_EDITION} \bowtie \text{ACHAT})$

- Explication :

- Nous utilisons une jointure gauche entre MAISON_EDITION et ACHAT pour inclure toutes les maisons, même si elles n'ont pas d'achat associé.
- La projection $\pi_{\pi\pi\pi}$ permet d'afficher uniquement le nom des maisons et la date d'achat si elle est présente.

ii) Listez les numéros des maisons d'édition dont la librairie a acheté tous les livres dont le prix de vente est de moins de 15€

- **Requête algébrique :**

$$(\pi_{NO_MAISON, NO_LIVRE}(ACHAT)) \div (\pi_{NO_LIVRE}(\sigma_{MNT_PRIX_VENTE_LIV < 15}(LIVRE)))$$

- **Explication :**

- La division \div est utilisée pour trouver les maisons d'édition ayant acheté tous les livres dont le prix de vente est inférieur à 15.
- La sous-requête sélectionne les NO_LIVRE des livres ayant un prix de vente inférieur à 15.

d) Agrégation

i) Comptez le nombre de livres achetés par chaque maison d'édition depuis le 12 décembre 2016

- **Requête algébrique :**

$$NO_MAISON \ \mathfrak{J}_{COUNT(NO_LIVRE)}(\sigma_{DATE_{ACH} > "2026-12-12"}(ACHAT))$$

- **Explication :**

- La sélection σ filtre les achats effectués depuis le 12 décembre 2016.
- Ensuite, nous effectuons une agrégation pour compter le nombre de livres (COUNT(NO_LIVRE)) achetés par chaque maison d'édition (NO_MAISON).