

TCH054 – SÉANCE 3

PROCESSUS DE CONCEPTION

LE MODÈLE RELATIONNEL

ÉTS 2

PLAN DE LA SÉANCE

- Qu'est-ce que le modèle logique ?
- Obtenir le modèle logique (relationnel) de données depuis le modèle conceptuel en quatre étapes:
 1. Identification des entités
 2. Identification des clés
 3. Transformation des relations (associations, généralisations)
 4. Normalisation

ÉTS
3

LE MODÈLE RELATIONNEL (ERD)

The diagram illustrates the relationship between three models in an Entity-Relationship (ER) diagram:

- Modèle conceptuel de données** (Conceptual Data Model): Represented by a grey rounded rectangle.
- Modèle logique (ou relationnel) (Logical Model)** (or Relational Model): Represented by a blue rounded rectangle.
- Modèle Physique (SQL) (Physical Model)** (SQL Model): Represented by a brown rounded rectangle.

Arrows indicate a flow from the conceptual model to the logical model, and from the logical model to the physical model. The conceptual model is also shown with a connection to a detailed ER diagram below.

Detailed ER Diagram:

- Author:** Entity with attributes: id (PK), nom (varchar20), prenom (varchar20).
- Book:** Entity with attributes: id (PK), titre (varchar20), auteur_id (FK), etat (enum: bon, moyen, mauvais), nb_pages (number(10)), etat (enum: bon, moyen, mauvais), etat (enum: bon, moyen, mauvais).
- Client:** Entity with attributes: id (PK), nom (varchar20), prenom (varchar20), adresse (varchar21250).
- Genre:** Entity with attributes: id (PK), nom (varchar20).
- Section:** Entity with attributes: id (PK), nom (varchar2100), non (varchar2100).
- Link:** Association with attributes: id (PK), auteur_id (FK), book_id (FK), etat (enum: bon, moyen, mauvais).
- Log:** Entity with attributes: id (PK), date (date), heure (time), client_id (FK), etat (enum: bon, moyen, mauvais).

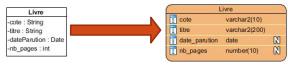
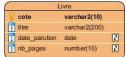
ÉTS
4

TRANSFORMATION DU MODÈLE RELATIONNEL EN MODÈLE LOGIQUE

- Durant l'élaboration du modèle conceptuel nous avons identifiés les principaux concepts (classes) du domaine d'affaires.
- Cette analyse nous servira directement : le modèle relationnel découle du modèle conceptuel
- Le modèle conceptuel sera étendu par des détails spécifiques à la persistance dans une base de données relationnelle.

ÉTS
5

OBTENTION DU MODÈLE RELATIONNEL EN 4 ÉTAPES:

- Identifier les entités: ce sont les classes du modèle conceptuel
 
- Identifier les clés primaires et les contraintes statiques
 
- Traduire les associations
- Normaliser

ÉTS
6

6

ÉTAPE 1: DÉFINIR DES ENTITÉS

ÉTS

1

ENTITÉ

- Un ensemble homogène d'informations qui correspondent au même "objet" à faire persister dans la base de données.
- S'apparente à la notion de classe - une variation terminologique:
 - Classe → Entité
 - Attribut → Colonne

ÉTS

8

CLASSE → ENTITÉ

- On commence par créer une entité pour chacune des classes du modèle conceptuel
- Les attributs de la classe deviennent des colonnes de l'entité

Livre - <i>cote</i> : String - <i>titre</i> : String - <i>dateParution</i> : Date - <i>nb_pages</i> : int	 Livre <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td><input type="checkbox"/></td> <td><i>cote</i></td> <td>varchar2(10)</td> </tr> <tr> <td><input type="checkbox"/></td> <td><i>titre</i></td> <td>varchar2(200)</td> </tr> <tr> <td><input type="checkbox"/></td> <td><i>date_parution</i></td> <td>date</td> </tr> <tr> <td><input type="checkbox"/></td> <td><i>nb_pages</i></td> <td>number(10)</td> </tr> </table>	<input type="checkbox"/>	<i>cote</i>	varchar2(10)	<input type="checkbox"/>	<i>titre</i>	varchar2(200)	<input type="checkbox"/>	<i>date_parution</i>	date	<input type="checkbox"/>	<i>nb_pages</i>	number(10)
<input type="checkbox"/>	<i>cote</i>	varchar2(10)											
<input type="checkbox"/>	<i>titre</i>	varchar2(200)											
<input type="checkbox"/>	<i>date_parution</i>	date											
<input type="checkbox"/>	<i>nb_pages</i>	number(10)											

ÉTS

9

PRINCIPAUX TYPES DE DONNÉES

Modèle conceptuel	Modèle logique
Int	Number(p)
double	Number(p,s)
String	Varchar(n)
Date	Date
Enum	<ul style="list-style-type: none"> - Si nombre de valeurs limité: Varchar(n) avec contrainte - Si nombre de valeurs important (ou non connu d'avance): Crée une autre entité

Le type **Number**:

- p (precision): nombre total de chiffres (incluant décimales)
- s (scale): nombre de décimales

ÉTS
10

COLONNES NULLABLES

- Une colonne est **nullable** s'il est permis qu'elle n'ait pas de valeurs pour un ou des enregistrements.
- Un enregistrement n'ayant pas de valeur pour une colonne donnée recevra la valeur **NULL**.
- Nous devons identifier les colonnes où les valeurs **NULL** sont permises
→ Pour toutes les autres colonnes, les valeurs nulles ne sont pas permises

Live

cote	varchar2(10)	
titre	varchar2(200)	
date_parution	date	N
nb_pages	number10	N

ÉTS
11

CONVENTIONS DE NOMMAGE

- On utilise la convention "snail_case" pour nommer autant les entités que les attributs.
- Les mots d'une **entité** commencent par une majuscule. Ex.: **Mode_Transport**
- Les mots d'une **colonne** commencent par une minuscule. Ex.: **num_assurance_sociale**

ÉTS
12

EXERCICE



- Identifiez les entités du modèle conceptuel suivant en précisant les colonnes nullables.

Modèle Conceptuel (UML Class Diagram):

```

classDiagram
    class Adresse {
        -num : int
        -rue : String
        -ville : String
        -codePostal : String
    }
    class Client {
        -nom : String
        -prenom : String
        -numTel : String
    }
    class Transport {
        *-poids : double
        *-volume : double
        -adresse : String
        -adresseArrivee : String
        -distance : double
        -dateDebut : Date
        -dateFin : Date
    }
    class Camion {
        -plaque : String
        -capacite : double
    }
    Adresse "1" -- "1" Client : contrats
    Client "*" -- "*" Transport : livraisons
    Client "*" -- "1..*" Camion : ?
    Transport "*" -- "*" Camion : ?
  
```

ÉTAPE 2: IDENTIFIER LES CLÉS

13

ÉTS

13

QU'EST QU'UNE CLÉ?

- Clé:** Ensemble de colonnes d'une entité dont les valeurs combinées permettent d'identifier **de façon unique** une occurrence de l'entité.
- Exemple:** Toute salle est identifiée de façon unique par la combinaison de son numéro (ex.: 45) et l'étage où elle se trouve (ex.: 1)

Salle		
etage	number(2)	N
num_salle	number(3)	N
salle_info	char(1)	N
fenetres	char(10)	N

ÉTS

14

CLÉ CANDIDATE

- Toute clé au sein d'une entité est appelée clé candidate (candidate à être une clé primaire)

L'entité « Livre » comporte quatre clés candidates:

- id_livre
- cote
- code_isbn
- code_isbn_9

Livre		
id_livre	number(10)	
cote	varchar2(10)	
titre	varchar2(200)	
date_parution	date	N
nb_pages	number(10)	
code_isbn	varchar2(12)	
code_isbn_9	varchar2(9)	

- Note:** une clé candidate ne peut **pas** être nullable.

ÉTS

15

ETS

16

CLÉ PRIMAIRE

- Une clé primaire est la clé candidate choisie pour identifier de façon unique chaque enregistrement dans la base de données.
- Toute entité doit obligatoirement avoir une et une seule clé primaire.
- Le choix de la clé primaire dépend:
 - Des règles d'affaire (utilisée naturellement dans le domaine)
 - Des préoccupations de performances (limiter la taille des index)
 - Des aspects de normalisation (éviter les clés composées)
- Si la clé primaire provient des colonnes existantes, c'est une clé « naturelle »

id_avis	number(10)
code	varchar(20)
titre	varchar(200)
date_publication	date
nb_pages	number(10)
code_avis	varchar(2)
code_avis_3	varchar(20)

ETS

17

CLÉ NATURELLE/ARTIFICIELLE

- En l'absence de clé candidate, nous sommes dans l'obligation de « fabriquer » une clé primaire: **une clé artificielle**
- C'est une clé (généralement numérique) qui ne provient pas des données naturelles de l'entité

Client	
id_client	number(10)
nom	varchar(250)
prenom	varchar(250)
telephone	varchar(250)

- Nous pouvons créer des clés artificielles aux fins d'amélioration de la performance ou de normalisation.

ETS

18

MODÈLE LOGIQUE

- Pour toute entité du modèle logique obtenue à l'étape 1, nous devons:
 1. Identifier les clés candidates
 2. Choisir la clé primaire.

Si aucune clé primaire → créer une clé numérique artificielle

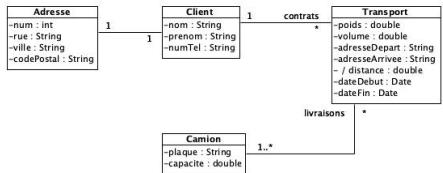
 3. Pour les autres clés candidates (non nullables!!), définir une contrainte d'unicité.
 4. En l'absence de clé candidate, ou face à des clés composées, créer une clé numérique artificielle.

CONVENTIONS DE NOMMAGE

- **Une clé primaire naturelle** garde son nom normal. Ex.:
matricule
 - **Une clé primaire artificielle** respecte le format nom:
id_nom_table
Exemple, la clé artificielle de la table livre: **id_livre**.
 - **Une clé étrangère** : Dépend de la clé primaire à laquelle on fait référence
 - Si c'est une clé artificielle: la clé étrangère aura le même nom que la clé primaire
 - Sauf si c'est une relation réflexive, on utilise les noms des rôles.
 - Si c'est une clé naturelle: On lui ajoute le nom de la table d'où provient la clé primaire. Exemple:
employe_num_ass_sociale



- Continuez l'exercice en identifiant les clés primaires



ETS

22

LES RELATIONS DANS LE MODÈLE RELATIONNEL

- Le modèle relationnel ne supporte que des associations de type:
 - 1 - 1 (0 - 1)
 - 1 - n (0 - n)
 - Dans le modèle conceptuel, nous avons pu représenter plusieurs types de relations:
 - Des associations de cardinalité multiples
 - Des relations de généralisations
 - D'autres types de relations non abordées.
- Il faut traduire les relations du modèle conceptuel en se limitant aux relations de type 1-1 et 1-N.

ETS

23

LES CLÉS ÉTRANGÈRES

- Une clé étrangère est une/des colonnes d'une entité A, qui fait référence à une clé primaire d'une autre entité B.
- Concrètement, ceci permet de réaliser la liaison entre deux entités.

Clé primaire	CodePerm	Nom	Prenom	DateNaiss
	SNOJ1982102101	Snow	Jon	1982-10-21
	STAA1987071509	Stark	Arya	1987-01-10
	TARD1979071502	Targaryen	Daenerys	1979-07-15
Clé étrangère	CodePerm	Epreuve	Note	
	SNOJ1982102101	Intral	85	
	STAA1987071509	Intral	92	
	TARD1979071502	Intral	65	

ETS

24

RELATIONS 1-1, 1-N ET M-N

- Pour traduire une association du modèle conceptuel, on se base sur la cardinalité de l'association:
- Si la cardinalité permet un nombre d'éléments associés supérieur à 1, c'est une relation de type **plusieurs**



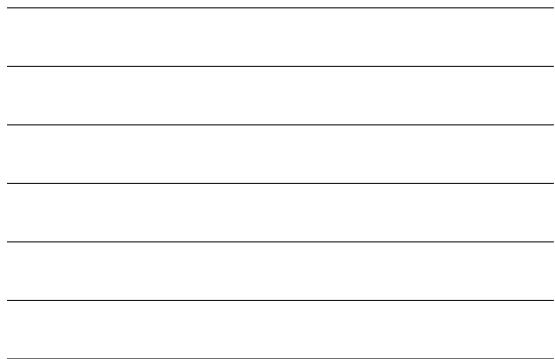
- Exemples:
 - 0..5
 - 3..10
 - 2..*
 - 1..*
 - *

RELATIONS 1-1, 1-N ET M-N

- Si la cardinalité permet un nombre d'éléments associés inférieur ou égal à 1, c'est une relation de type **un**



- Exemples:
 - 0..1
 - 1



RELATIONS 1-1, 1-N ET M-N

- Nous cherchons à définir le type de relation qui relie deux classes:

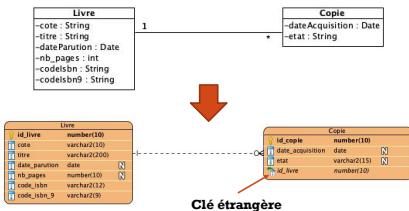
- Relation un-à-un (1-1)
 - Relation un-à-plusieurs (1-N)
 - Relation plusieurs-à-plusieurs (N-N)

- **Exemple:** Une relation 1-N



TRANSFORMATION DES ASSOCIATIONS 1 - N

- #### ▪ Placer une clé étrangère du côté plusieurs (N)



TRANSFORMATION DES ASSOCIATIONS

1 - N

Table: Livre

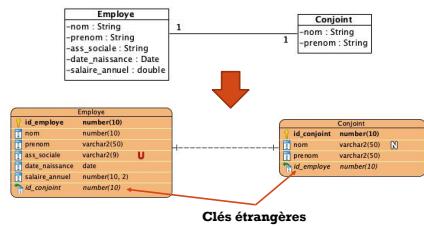
id_livre	titre	...
1	Century Trilogy	
2	Kingsbridge	
3	Les misérables	
4	La communauté de l'anneau	

Table: Copie			
id_copie	etat	...	Id_livre
1	Parfait		4
2	Moyen		4
3	Parfait		2
4	Mauvais		1

TRANSFORMATION DES ASSOCIATIONS

1 - 1

- Option 1: Placer une clé étrangère dans une des deux entités OU les deux (selon la navigabilité)



TRANSFORMATION DES ASSOCIATIONS

1 - 1

Table-Livre

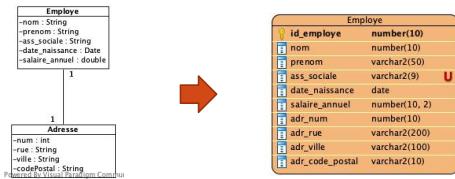
Table: Invac				
id_emp	nom	prenom	...	id_conj
10	Tremblay	Julie		32
15	Lacasse	Annie		27
29	Labonté	Steve		23
44	Ouellet	Mike		13

Table: Copie				
id_conj	nom	...	id_employee	
13	Conjoint1		44	
23	Conjoint2		29	
27	Conjoint3		15	
32	Conjoint4		10	

TRANSFORMATION DES ASSOCIATIONS

1 - 1

- Option 2: Fusionner les entités



- Quand favoriser l'option 1 ou l'option 2?

ÉTS

TRANSFORMATION DES ASSOCIATIONS

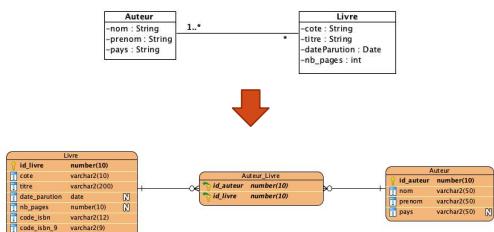
M - N

- En relationnel, il n'est pas possible de modéliser des relations M-N.
 - On crée une table additionnelle (appelée **table de jointure** ou **table d'intersection**) qui représente la relation.
 - On obtient deux relations de type 1-N (qu'on sait représenter)
 - La table de jointure contient les clés étrangères des deux autres entités.
 - Les deux clés étrangères de la table de jointure constituent sa clé primaire.

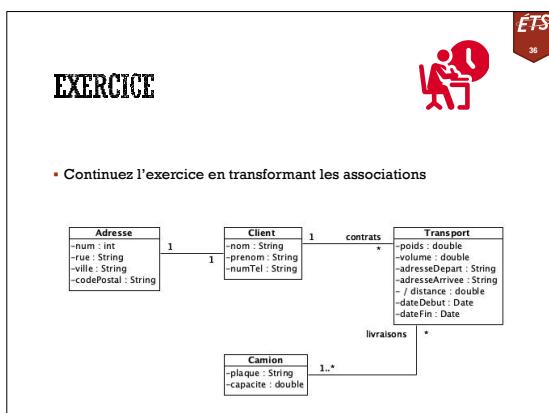
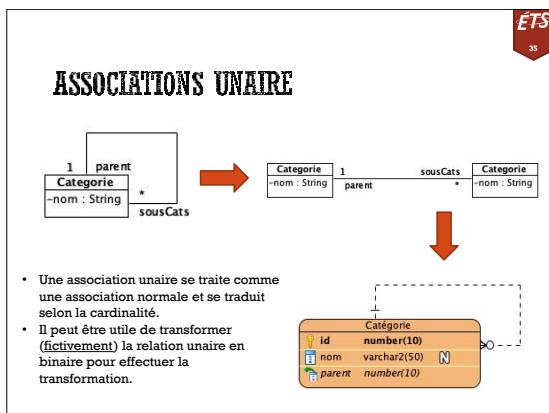
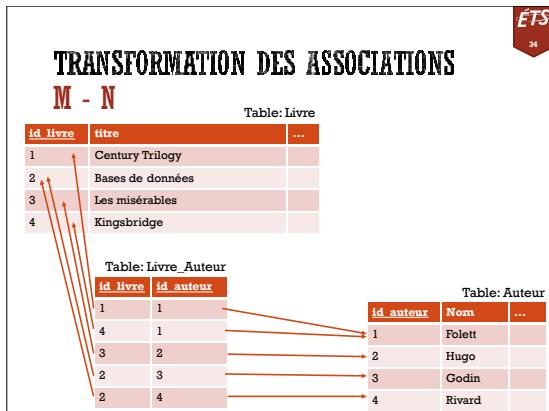
ÉTS

TRANSFORMATION DES ASSOCIATIONS

M - N



ÉTS

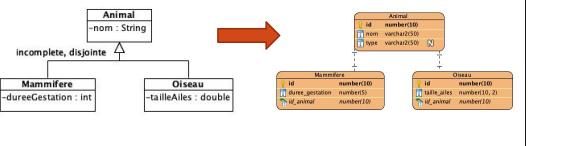


GÉNÉRALISATIONS

- La notion de généralisation/héritage n'existe pas dans les bases de données relationnelles.
- Cependant ce concept existe dans les langages de programmation
- Nous allons transformer les relations en:
 - Fusonnant des tables
 - Utilisant des relations de type 1-1.
- L'approche dépend du type de relation (disjointe/chevauchante, complète/incomplète).
- Cf. Godin R., 2006.

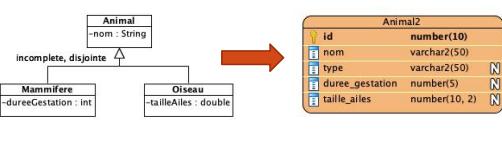
GÉNÉRALISATION - INCOMPLÈTE

- Si la relation est de type **incomplète**:
 - Il faut maintenir la table générale
 - Ajouter un discriminant à la table générale
 - Les tables spécialisées peuvent être:
 - Maintenues avec une relation 1-1 avec la table générale



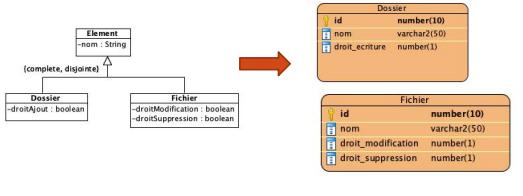
GÉNÉRALISATION - INCOMPLÈTE

- Si la relation est de type **incomplète**:
 - Il faut maintenir la table générale
 - Ajouter un discriminant à la table générale
 - Les tables spécialisées peuvent être:
 - Concaténées à la table générale (concaténation vers le haut) mais provoque l'ajout de beaucoup de colonnes nullables



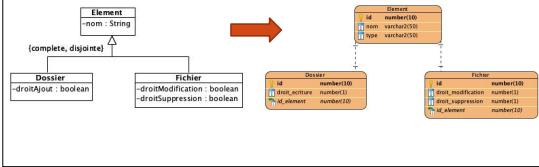
GÉNÉRALISATION - COMPLÈTE

- Si la relation est de type **complète**:
 - Les tables spécialisées sont maintenues
 - La table générale peut-être
 - Concaténée aux tables spécialisées (concaténation vers le bas)



GENERALISATION - COMPLÈTE

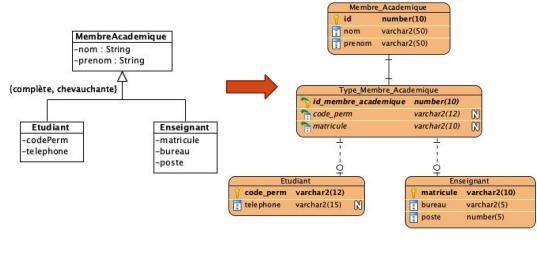
- Si la relation est de type **complète**:
 - Les tables spécialisées sont maintenues
 - La table générale peut-être
 - Maintenue avec des relations 1-1 avec les tables spécialisées et l'ajout d'un discriminant dans la table générale (si la table générale a des associations)



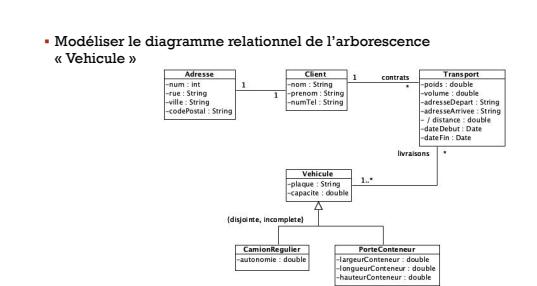
GÉNÉRALISATION — DISJOINTE/CHEVAUCHANTE

- Dans le cas d'une généralisation disjointe, on applique seulement les règles complète/incomplète.
 - Dans le cas d'une généralisation chevauchante:
 - la table générale ne doit pas être fusionnée: une entité est créée pour chacune des tables générale et spécialisées
 - Une table de jointure est ajoutée contenant des clés étrangères vers chacune des clés primaires de la généralisation
 - Les clés étrangères correspondantes aux clés primaires des tables spécialisées sont nullables.
 - La clé étrangère correspondante à la clé primaire de la table générale sera également la clé primaire de la table de jointure.

GENERALISATION – CHEVAUCHANTE



EXERCICE



ÉTAPE 4: NORMALISATION

QU'EST-CE QUE LA NORMALISATION?

- Approche permettant de résoudre des problèmes de conception menant à des redondances de données
 - Corollaire: Permet d'éviter les mises à jour incohérentes (anarchiques)
 - Plusieurs causes sont identifiées et sont résolues par la mise en forme normale de la table
 - Il existe plusieurs formes normales interdépendantes: 1FN, 2FN, 3FN, FNBC, 4FN, 5FN et d'autres

EXEMPLE DE PROBLÈME

- Considérons l'entité Vente, obtenue suite à la transformation du modèle conceptuel en modèle relationnel:

Vente	
no_commande	number(10)
date_commande	date
no_client	number(10)
nom_client	varchar(20)
no_telephone	varchar(20)
no_article	number(10)
description	varchar(256)
prix_unitaire	number(10, 2)
quantite	number(5)

EXEMPLE DE PROBLÈME

Redondance de données

Source: R. Godin, 2013

SOLUTION: DÉCOMPOSITION

TABLE VENTE									
NOCOMMAND	DATECOMMAND	NOCLIENT	DOMAILLE	DOMAILLE	POTELLEUR	NOARTICLE	DESCRIPTION	PIEUXUNITAIRE	QUANTITE
1	08/06/2000	100	T. U. 7	1999/9999-9999	00	1	CLIQUE EN	10-99	10
			SANSIMAI						
1	08/06/2000	1	T. U. 7	1999/9999-9999	70	1	CLIQUE EN A	10-99	5
			SANSIMAI						
1	08/06/2000	10	T. U. 7	1999/9999-9999	90	1	PO NUMBER	11-99	1
			SANSIMAI						

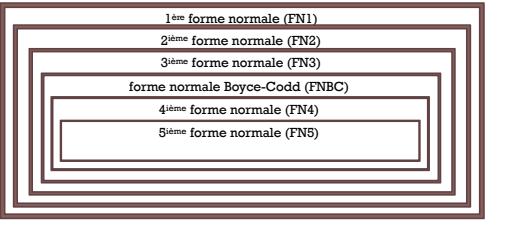
TABLE CLIENT									
NOCLIENT	DOMAILLE	NOTELLEUR	NOVTELEUR						
1	T. U. 7	1999/9999-9999	00						
2	O. L. L. A. R.	5883888-8888							
3	PREMISLAY								
4	21N8	77777777							
5	FRÉDÉRIC LAROCHE	66666666							
6	FRÉDÉRIC LAROCHE	55535555-5555							
7	DOMITÈSE HAKER	6668666-6666							
8	ZOU COUPUREUR	4441444-1449							
9	W. L. L. A. R.	13551355-1351							
10	PREMISLAY								

TABLE VENTE									
NOCOMMAND	DATECOMMAND	NOCLIENT	DOMAILLE	DOMAILLE	POTELLEUR	NOARTICLE	DESCRIPTION	PIEUXUNITAIRE	QUANTITE
1	08/06/2000	100	T. U. 7	1999/9999-9999	00	1	CLIQUE EN	10-99	10
			SANSIMAI						
1	08/06/2000	1	T. U. 7	1999/9999-9999	70	1	CLIQUE EN A	10-99	5
			SANSIMAI						
1	08/06/2000	10	T. U. 7	1999/9999-9999	90	1	PO NUMBER	11-99	1
			SANSIMAI						

Source: R. Godin, 2013

LES FORMES NORMALES

- Une table est dans une forme normale donnée si elle satisfait à certaines propriétés de normalisation



PREMIÈRE FORME NORMALE (FN1)

- Une table est en FN1 si **elle ne contient aucune colonne multivaluée** (tous les attributs sont atomiques)
 - Une colonne est multivaluée si elle contient plus d'une valeur

The following table illustrates the data content plus a few variations.		
no_emp	telephone	diplomes
15	514-784-8899	BACC info, MSc finance
16	450-784-7788	BACC ELE
17	418-895-1122	BACC MEC, MSc Gestion

no_emp	telephone	diplome
15	514-784-8899	BACC info
15	514-784-8899	MSc finance
16	450-784-7788	BACC ELE
17	418-895-1122	BACC MEC
17	418-895-1122	MSc Gestion

DEUXIÈME FORME NORMALE (FN2)

- Une table respecte la 2^e forme normale si:
 - Elle respecte la FN1
 - Tout attribut ne faisant pas partie de la clé est complètement dépendant de la clé primaire (ne dépend pas d'une partie de la clé)
 - Cette forme normale ne s'applique que dans le cas de clés composées (multi-colonnes)

ÉTS
52

DEUXIÈME FORME NORMALE (FN2)

- Exemple non FN2

Souligné : Clé

Table: Employe_Projet				
<u>id_employe</u>	<u>id_projet</u>	role	nom_employe	nom_projet
10	45	Programmeur	Lapierre, A.	Caisse de dépôt
12	37	Designer FrontEnd	Desjardins, R.	Therac
10	33	Architecte	Lapierre, A.	Geothermia
18	37	Concepteur	Labonté, T.	Therac

Source: Cours GPA775 (ÉTS)

ÉTS
53

DEUXIÈME FORME NORMALE (FN2)

- Table normalisée en FN2

Table: Employe		Table: Projet	
<u>id_employee</u>	nom	<u>id_projet</u>	nom
10	Lapierre, A.	45	Caisse de dépôt
12	Desjardins, R.	33	Geothermia
18	Labonté, T.	37	Therac

Table: Employe_Projet		
id_employe	id_projet	role
10	45	Programmeur
12	37	Designer FrontEnd
10	33	Architecte
18	37	Concepteur

Source: Cours GPA775 (ETS)

54

TROISIÈME FORME NORMALE (FN3)

- Une table est en troisième forme normale si:
 - Elle respecte la FN2
 - Toute colonne non-clé dépend directement (non-transitivement) de la clé primaire.
(i.e. ne dépend pas d'un ou plusieurs attributs n'appartenant pas à la clé).
- Exemple:

Table: Etudiant			
IdEtudiant	NomEtudiant	Département	SigleDep
3	Bernard	115	MEC
5	Labbé	115	MEC
8	Barrette	119	GPA
9	Cadieux	117	ELE

Source: Cours GPA775 (ETS)

TROISIÈME FORME NORMALE (FN3)

- Table normalisée en FN3

Table: Etudiant		
IdEtudiant	NomEtudiant	Département
3	Bernard	115
5	Labbé	115
8	Barrette	119
9	Cadieux	117

Table: Département	
IdDépartement	Sigle
115	MEC
119	GPA
117	ELE

Source: Cours GPA775 (ETS)

FORME NORMALE BOYCE-CODD (FNBC)

- Une table est en forme normale Boyce-Codd (FNBC) si:
 - Elle respecte la FN3
 - Tous les attributs non-clé ne sont pas source de dépendance vers une partie de la clé.

Table: Etudiant_Cours			
IdEtudiant	IdCours	IdEnseignant	CoteObtenue
3	12	1234	B
5	15	6789	A-
8	12	1234	B+
5	17	1234	A

Source: Cours GPA775 (ETS)

ÉTS
58

FORME NORMALE BOYCE-CODD (FNBC)

- Table normalisée en FNBC:

Table: Etudiant_Cours

IdÉtudiant	IdCours	CoteObtenue
3	12	B
5	15	A-
8	12	B+
5	17	A

Table: Cours

IdCours	IdEnseignant
12	1234
15	6789
17	1234

Source: Cours GPA775 (ÉTS)

ÉTS
59

QUATRIÈME FORME NORMALE (FN4)

- Une table est en 4^{ème} forme normale si:
 - Elle respecte FNBC
 - Ne contient pas de dépendances multivaluées.

- Exemple de violation de FN4

Table: Cours_EsEnseignant_Livre

cours	enseignant	livre_reference
Analyse	Lessage	Livre a
Analyse	Jacob	Livre b
Analyse	Desjardins	-
Program. Mobile	Saulnier	Livre d
Program. Mobile	Leblanc	Livre e

- cours →→ enseignant et cours →→ livre_reference
- Aucune dépendance entre enseignant et livre_reference

ÉTS
60

QUATRIÈME FORME NORMALE (FN4)

- Normalisation en FN4:

Table: Cours_EsEnseignant

cours	enseignant
Analyse	Lessage
Analyse	Desjardins
Program. Mobile	Saulnier

Table: Cours_Reference

cours	livre_reference
Analyse	Livre a
Analyse	Livre b
Program. Mobile	Livre d
Program. Mobile	Livre e

ÉTS
61

FORMES NORMALES – RÉSUMÉ

- **FN1:** Pas d'attributs multivalués
- **FN2:** FN1 + Pas de dépendances partielles à la clé
- **FN3:** FN2 + Pas de dépendances vers une colonne non-clé
- **FNBC:** FN3 + Une colonne non-clé n'introduit pas de dépendances vers une partie de la clé
- **FN4:** FNBC + Pas de dépendances multivaluées
- **FN5** (non traitée): FN4 + Toute dépendance de jointure dans la table découle des clés candidates de la table

ÉTS
62

RECETTE POUR UNE NORMALISATION FACILE

- Il est généralement assez aisé de créer des tables normalisées en 4NF en respectant certaines règles simples:
 - Créer des tables à thème unique.
Ex.: une table Client ne devrait pas avoir de colonnes autres que des colonnes qui décrivent directement un client (i.e. pas les factures, etc.)
 - Éviter des clés primaires composées et favoriser des clés artificielles numériques
 - Éviter toute forme de dépendances multivaluées : créer une nouvelle table qui contiendra uniquement chacun des champs multivalués

ÉTS
63

LES LIMITES DE LA NORMALISATION

- La normalisation permet d'éviter certaines anomalies et des redondances
- La normalisation introduit une complexité au niveau du modèle relationnel à travers les décompositions successives et des associations introduites entre les tables.
- La normalisation n'est pas une fin en soi: On ne vise pas toujours absolument FN6 – On peut accepter (et tenir compte) de certaines anomalies au profit de la performance
→ Dénormalisation...
- Cependant, dans la majorité des cas, les tables devraient être au moins en FNBC.

PROCESSUS DE CONCEPTION D'UNE BASE DE DONNÉES RELATIONNELLE

