

Cours de
Base de données



Normalisation & Conversion



Plan

01

**Modèle conceptuel
Modèle relationnel**

02

**Règles de passage du modèle
conceptuel au modèle relationnel**

03

**Exemples de
redondances à éliminer**

04

Normalisation : 3 règles

MODÈLES & CONVERSION

1

2



Modèle conceptuel

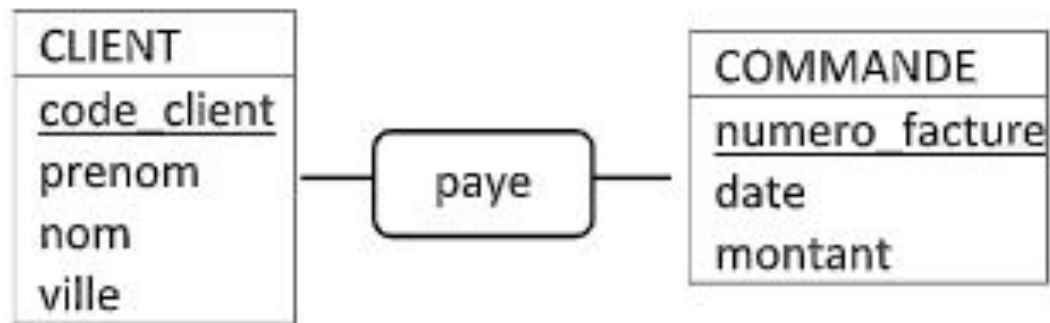
Le modèle conceptuel se réalise à l'étape fonctionnelle d'un projet. L'étape fonctionnelle c'est le moment dans lequel on analyse les besoins et décide ce que sera le logiciel.

Cette technique consiste à énumérer, d'une manière relâchée, les concepts traités dans le logiciels et les champs, sans leur attribuer de type précis.

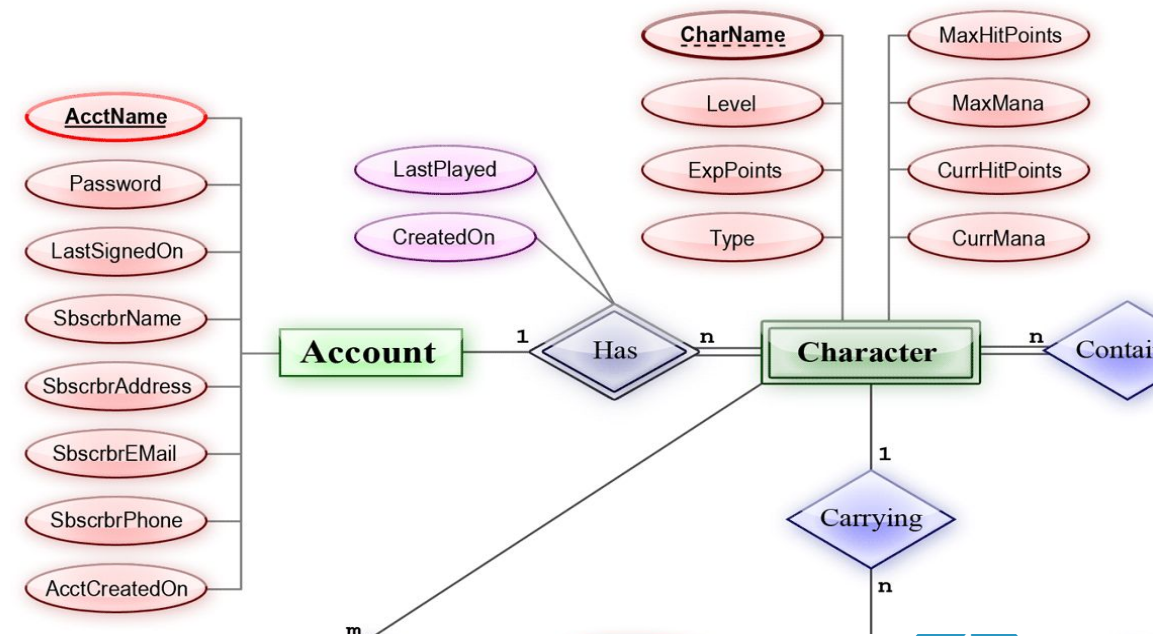
On identifie la clé primaire.

On trace des liens entre les données et on place des mots pour qualifier ces liens, sans préciser si ces liens sont des clés étrangères ou des tables de correspondance.

MCD = ERD



OU





Modèle relationnel

Le modèle relationnel constitue une **définition exacte** des tables et de leurs champs. Il peut définir ou non les types de données.

Les tables peuvent se présenter sous forme de tableau ou encore sous forme de description textuelle sous forme de RELATIONS.

MLD = relational schema

FACTURE(numero_facture, date, montant, code_client)
PRODUIT(slug_produit, nom, description, prix, photo)
ACHAT(numero_facture, slug_produit, nombre)

OU

FACTURE

numero_facture:string
date: timestamp
montant:float
code_client:string

ACHAT

numero_facture:string
slug_produit:string
nombre:integer

PRODUIT

slug_produit:string
nom:string
description:string
prix:float
photo:string

Exemple d'utilisation de la **relation** FACTURE

numero_facture	date	montant	code_client
1	2020-05-01	50.95	NG33
2	2020-05-01	100.67	SA21
5	2020-05-01	20.21	PO44
7	2020-05-01	50.11	LO42



Conversion du modèle conceptuel à relationnel

Règle 1 - Toute entité devient une relation ayant pour clé primaire son identifiant.

Chaque propriété se transforme en attribut

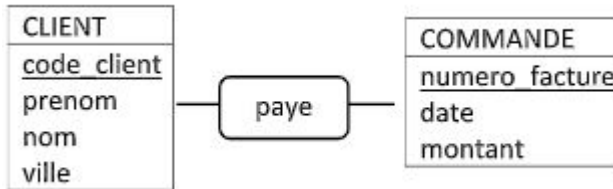
CLIENT
<u>code_client</u>
prenom
nom
ville



CLIENT(code_client,prenom,nom,ville)

Règle 2 - Toute association hiérarchique (de type [1, n]) se traduit par une clé étrangère.

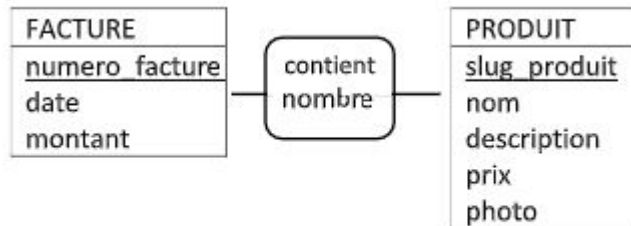
La clé primaire correspondant à l'entité parent (côté n) migre comme clé étrangère dans la relation correspondant à l'entité fils (côté 1)



CLIENT(code_client,prenom,nom,ville)
FACTURE(numero_facture,date,montant,
code_client)

Règle 3 - Toute association non hiérarchique (de type [n, n]) devient une nouvelle relation

La clé primaire est formée par la concaténation (juxtaposition) l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles deviennent des attributs qui ne peuvent pas faire partie de la clé.



FACTURE(numero_facture,date,montant,
code_client)
PRODUIT(slug_produit,nom,description,prix,
photo)
ACHAT(numero_facture,slug_produit,nombre)

REDONDANCE & NORMALISATION

1

2

3

3 règles.



Principe de la NORMALISATION

La NORMALISATION consiste à appliquer des règles (les FORMES NORMALES) pour corriger des anomalies ou des redondances dans le modèle d'une base de données. Son interprétation exacte dépend de la nature des données et aussi de leur utilisation dans l'application.

On peut l'appliquer sur l'étape du modèle conceptuel (MCD - DER).

On peut l'appliquer sur le modèle relationnel des données (TABLES)



NORMALISATION

01

1NF - Première forme normale

02

2NF - Seconde forme normale

03

3NF - Troisième forme normale

1NF - 1ère règle normale

**BONNE
PRATIQUE**

Une relation est normalisée en première forme normale SI :

- Elle possède une clé identifiant de manière unique et stable chaque ligne
- Chaque attribut est monovalué (ne peut avoir qu'une seule valeur par case)
- Aucun attribut n'est décomposable en plusieurs attributs significatifs
 - Cette dernière règle est subjective et dépend de l'application programmée. Par exemple l'adresse peut être considérée comme décomposable ou non-décomposable selon le cas.

<u>id</u> (clé primaire)	matricule	prenom	nom	id_programme
1	9785421	Jean	Desbiens	1
2	9785622	Julia	Bilodeau	1
5	9845622	Francis	Lavoie	2
7	9738300	Marion	Boisvert	2

1NF - CONTRE-EXEMPLE - table etudiant

CONTRE
EXEMPLE

La clé n'est pas
stable, elle n'est
pas contrôlée
par la base de
donnée

Aucun attribut
n'est
décomposable

Chaque
attribut est
monovalué

<u>matricule</u>	prenom & nom	id_programme	liste_des_cours
INFO9785421	Jean Desbiers	1	Architecture, Programmation mobile, Portefolio
INFO9785622	Julia P. deau	1	Architecture, Programmation mobile, Portefolio
ELEC9845622	Francis Lavoie	2	Mathématiques, Électronique 1, Français 1
ELEC9738300	Marion Boisvert	2	Mathématiques, Électronique 1, Français 1

De plus, les
programmeurs
préfèrent avoir
des numéros
comme id

SIGNIFICATIF
car on peut
vouloir trier par
nom de famille

Si on veut faire
des listes
d'étudiants par
cours ?

2NF - 2ème règle normale

BONNE
PRATIQUE

Une relation est normalisée en deuxième forme normale si :

- Elle est en première forme normale
- Tout attribut non-clé est dépendant de la clé au complet

La 2NF est à vérifier seulement pour les relations avec une clé composée

<u>id</u> (clé primaire)	no_employe	prenom	nom	bureau
1	2288	Nadine	Giasson	O-131
2	3177	Francis	Tremblay	O-149
5	4499	Francis	Haché	J-170
7	1100	Nadine	Rouleau	J-111

2NF - CONTRE-EXEMPLE - table professeur

CONTRE
EXEMPLE

2NF n'est pas une règle importante à analyser car elle est automatiquement respectée en utilisant les id auto_increment et serial

Si le casier dépend du nom seulement, il est enfreint 2NF

<u>prenom</u> (1ère partie clé)	<u>nom</u> (2ème partie clé)	casier	bureau
Nadine	Giasson	G	O-131
Francis	Tremblay	T	O-149
Francis	Haché	H	J-170
Nadine	Rouleau	R	J-11

Savoir juste le prénom ne dit pas le bureau

3NF - 3ème règle normale

BONNE
PRATIQUE

Une relation est normalisée en troisième forme normale si :

- Elle est en seconde forme normale
- Tout attribut non-clé doit dépendre directement de la clé, c'est à dire qu'il ne doit pas dépendre par transitivité.

Une autre manière de le formuler est que l'attribut non-clé ne doit pas dépendre d'un autre attribut non-clé.

<u>id</u> (clé primaire)	no_employe	prenom	nom	departement	aile
1	2288	Nadine	Giasson	informatique	B
2	3177	Francis	Tremblay	informatique	B
5	4499	Francis	Haché	electronique	J
7	1100	Nadine	Rouleau	electronique	J

3NF - CONTRE-EXEMPLE - table professeur

Si l'aile dépend
du département,
cela devrait être
dans une table
département

Le nombre de
cours dépend des
données dans la
table cours

<u>id</u>	no_employe	prenom	nom	departement	aile	nombre_cours
1	2288	Nadine	Giasson	informatique	B	3
2	3177	Francis	Tremblay	informatique		2
5	4499	Francis	Haché	electronique	J	4
7	1100	Nadine	Rouleau	electronique	J	2

Plusieurs fois on
indique que le prof
d'électro est dans
l'aile J

Info
redondante
d'une ligne
à l'autre

On peut calculer le
nombre de cours
avec une jointure
de cours et la
fonction COUNT

Info
redondante
avec table
cours



MERCI