

INTRODUCTION AUX SYSTÈMES D'INFORMATION ET AUX BASES DE DONNÉES AVEC MYSQL

*

Objectifs

- Connaître les principales caractéristiques d'un système d'information
- Pouvoir participer à la modélisation d'une base de données
- Comprendre la structure d'une base de données
- Maîtriser les bases du langage SQL

Compléments nécessaires

- Accès à un serveur de bases de données (EasyPHP, MAMP, etc.)

Niveau de compétences en pré-requis

- Pas de connaissance particulières

*

*Ce document d'enseignement est diffusé librement, pour usage individuel.
Il est librement téléchargeable sur le site de l'auteur *.*

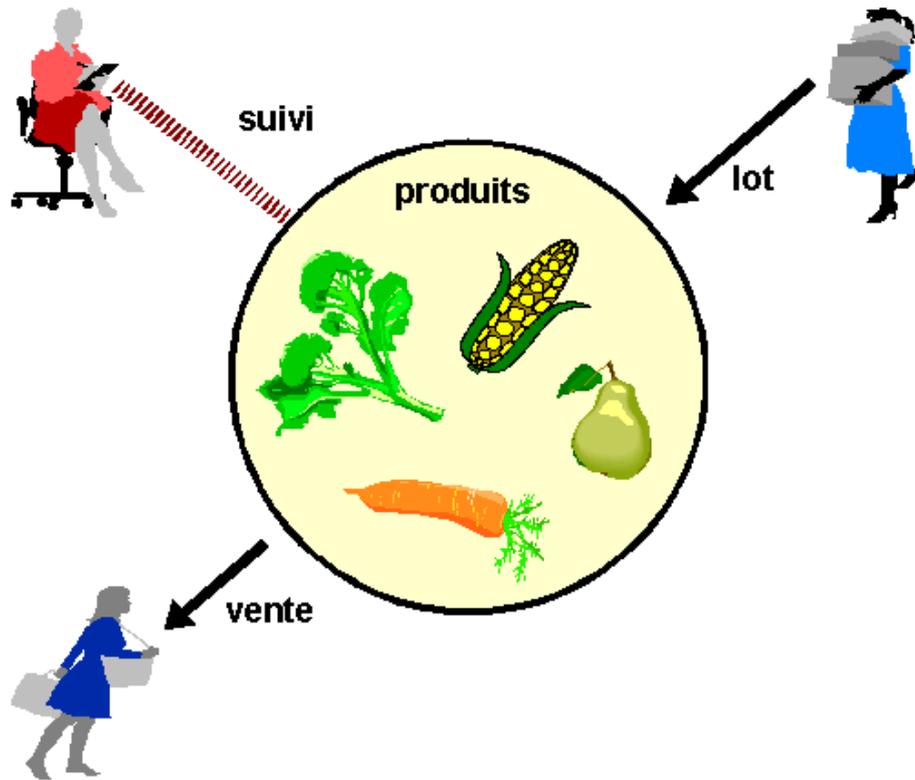
Michel Cartereau - Avril 2013

SYSTÈME D'INFORMATION

Introduction

Réalisation

INTRODUCTION AU SYSTÈME D'INFORMATION



● EXEMPLE D'ORGANISATION

- VENTE DE PRODUITS (FRUITS, LÉGUMES) PAR UNE COOPÉRATIVE
Produits achetés à un producteur et revendus à des clients
- DIFFÉRENTES ACTIVITÉS
Achat de lot à un producteur, vente de produit à un client,
suivi de l'évolution des stocks, etc.
- INFORMATIONS
Identification des produits avec prix de vente, d'achat et quantités,
coordonnées des producteurs et des acheteurs, etc.

● UN SYSTÈME D'INFORMATION

- UNE REPRÉSENTATION OPÉRATIONNELLE DE LA RÉALITÉ
Données (informations associées aux activités de l'organisation)
+
Outils (manipulation des données, aide à la décision)

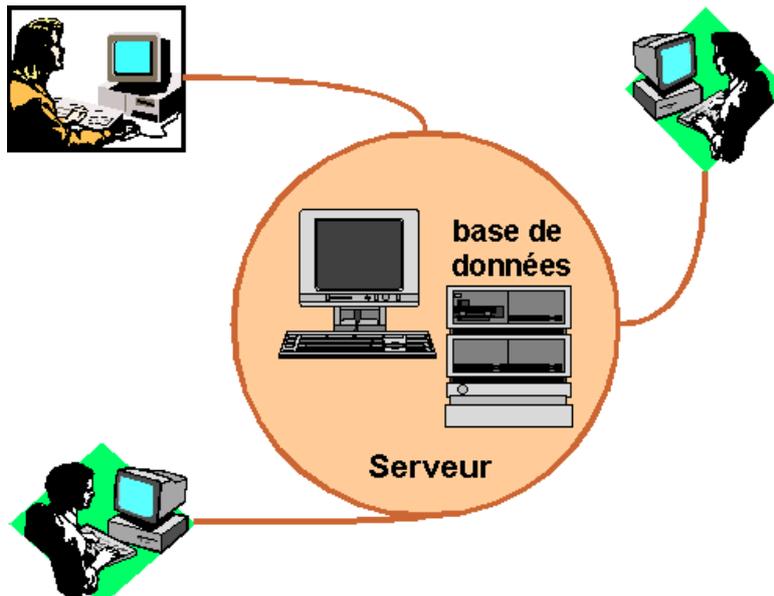
SYSTÈME D'INFORMATION : VUE INFORMATIQUE

QUELS SONT LES ÉLÉMENTS INFORMATIQUES ?

- UN SYSTÈME INFORMATIQUE

- UN SERVEUR

Une base de données + une application fournissant les outils



- DES POSTES « CLIENTS » DE TRAVAIL

Manipulations via des pages Web (mode d'accès quasi-universel)

 **La coopérative**
[Produits \(ajout\)](#) - [Personnes \(ajout\)](#) - [Lots \(ajout\)](#) - [Ventes \(ajout\)](#) - [Stock](#)

Bienvenue !

Michel Cartereau - AgroParisTech - Janvier 2009

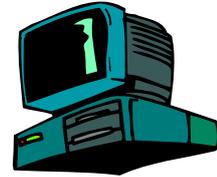
ARCHITECTURE CENTRALISÉE AVEC UTILISATIONS EN RÉSEAU

SYSTÈME D'INFORMATION : RÉALISATION

AUTOMATISATION DU FONCTIONNEMENT D'UNE ORGANISATION

● IMPORTANCE DU SYSTÈME D'INFORMATION

- ENJEUX CRITIQUES POUR L'ORGANISATION
Fonctionnement dépendant du système d'information
- UN OBJET POTENTIELLEMENT COMPLEXE ET COÛTEUX
Démarche méthodologique indispensable pour sa réalisation



● UNE DÉMARCHE DE PROJET

- GRANDES ÉTAPES
 - 1) Analyse des besoins, cahier des charges, étude de faisabilité
 - 2) Spécifications, conception et validation du système
 - 3) Mise en production et exploitation du système
- PRINCIPAUX RÔLES
Maîtrise d'ouvrage (MOA) → étapes 1 et 3
Maîtrise d'œuvre (MOE) → étape 2
- DIFFÉRENTES ACTEURS
Chefs de projet, architecte de logiciel, développeur, intégrateur, etc.



● RÔLE PRIMORDIAL DES DONNÉES

- LA BASE DE DONNÉES
Ensemble structuré d'informations cohérentes et pérennes, correspondant ici aux activités d'une organisation
- MODÉLISATION DES DONNÉES
Analyse des informations manipulées dans l'organisation avec représentation formelle de leur nature et leur structuration
- IMPLÉMENTATION DES DONNÉES
Utilisation d'un système de gestion de base de données (SGBD), et d'un langage de manipulation des données (norme SQL ¹)



FOCALISATION SUR LA BASE DE DONNÉES DANS CETTE PRÉSENTATION

¹ SQL : *structured query language*, langage de requêtes structuré

SYSTÈME D'INFORMATION : VUE PRATIQUE

A QUELLES SITUATIONS DE LA COOPÉRATIVE CELA CORRESPOND ?

● PRINCIPALES ACTIVITÉS

▪ DU CÔTÉ DES FOURNISSEURS

Enregistrement d'un nouveau lot, fournisseur ou produit

▪ DU CÔTÉ DES CLIENTS

Enregistrement d'une vente, d'un nouveau client,

▪ EN INTERNE

Mise à jour du prix d'un produit, rapports, état du stock, etc.



● SUIVI GÉNÉRAL

▪ EXEMPLES DE QUESTIONS

Quel est le prix du kilogramme de pomme ?

Quelle est l'adresse de De La Rue ?

Y-a-t-il des choux de Chine en stock ? Et combien ?

A qui correspond le numéro de téléphone 01.44.08.16.01 ?

Combien de légumes sont répertoriés ?

Quand est-ce que King King a livré des carottes ?

Combien de lots de brocolis a-t-on enregistré cette année ?

Quel est le poids moyen des ventes de carottes ?

Y-a-t-il un fruit dont le prix dépasse 10 euros ?

Quels sont les producteurs situés dans les Yvelines ?

Quel est le plus gros stock de produit actuel ?



LE SYSTÈME D'INFORMATION DOIT RÉPONDRE À TOUS LES BESOINS, QUI DOIVENT ÊTRE EXAMINÉS AVANT SA CRÉATION (« ANALYSE DES BESOINS »)

PRINCIPES DES BASES DE DONNÉES

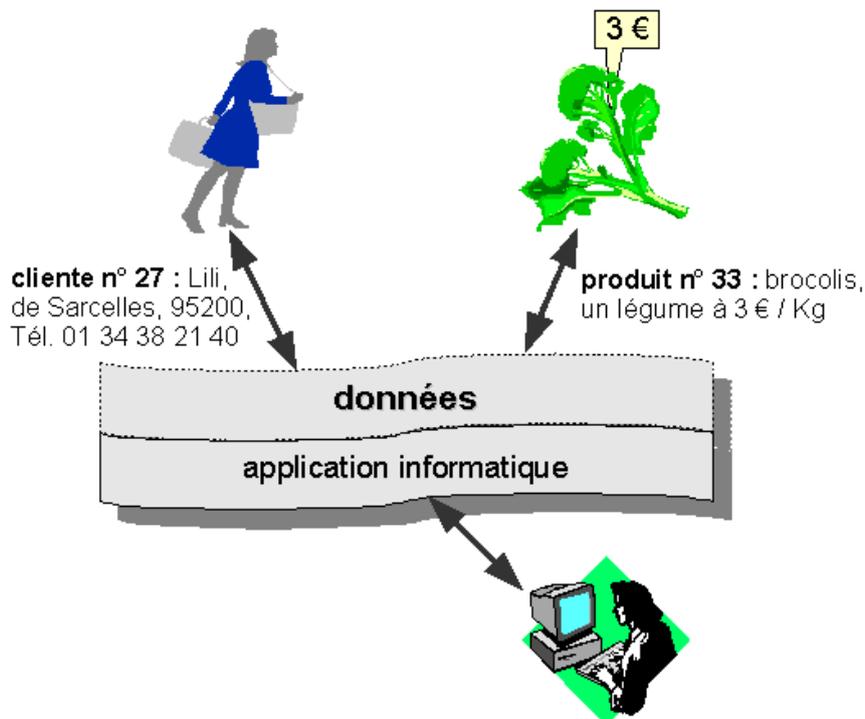
Modélisation des données

Modèle « entité-association »

Modèle « relationnel »

MODÉLISATION DES DONNÉES

QUELLES SONT LES INFORMATIONS MANIPULÉES ?



• MODÉLISATION CONCEPTUELLE

- DONNÉES : INFORMATIONS CORRESPONDANT À DES ÉLÉMENTS RÉELS
exemples : un client, un produit, un fournisseur, un lot, une vente
- INVENTAIRE DES ÉLÉMENTS RÉELS (« ENTITÉS »)
liste des données élémentaires ¹ rattachées (« attributs »)
exemple : produit avec numéro, nom, type (légume ou fruit), prix
- REPRÉSENTATION DE CHAQUE INFORMATION (« FORMAT »)
utilisation de types de données élémentaires (texte, nombre, etc.)
exemple : le nom d'un produit est un texte
- IDENTIFICATION DES DÉPENDANCES ENTRE LES ENTITÉS (« ASSOCIATIONS »)
exemple : une vente s'applique à un seul produit et un seul client,
avec une date et un prix associés

*LA MODÉLISATION DES DONNÉES INTERVIENT DANS LA PREMIÈRE ÉTAPE
DE LA CONCEPTION DU SYSTÈME D'INFORMATION
(MODÈLE CONCEPTUEL « ENTITÉ-ASSOCIATION » ² PRÉSENTÉ ICI)*

¹ Une donnée élémentaire ou « atomique », correspond à une valeur simple (nombre, texte, date, etc.) et non pas à une valeur multiple (liste, ensemble, etc.)

² En anglais : *entity-relationship model*, traduit aussi par modèle « entité-relation »

DONNÉE DU PRODUIT

MODÉLISATION DE L'ENTITÉ « PRODUIT » ¹



- COMMENT REPRÉSENTER UN PRODUIT ?

- DE QUOI A-T-ON BESOIN ?

Le nom du produit, exemple : carotte

Connaître son type, légume ou fruit, exemple : légume

Connaître son prix de vente (1 kilogramme), exemple : 1 euro

- COMMENT IDENTIFIER UN PRODUIT DE MANIÈRE UNIQUE ?

Identification possible avec son nom mais risque d'ambiguïté,
donc plus fiable avec un numéro d'ordre associé

- REPRÉSENTATION PAR LES ATTRIBUTS

1) **numéro** : nombre entier, l'« identificateur » ²

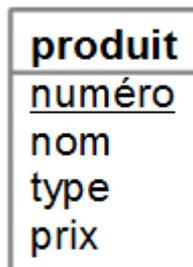
2) **nom** : texte

3) **type** : texte, soit LEGUME, soit FRUIT

4) **prix** : nombre décimal en euros, exemple : 1,5 (1 euro 50)

- MODÉLISATION

Représentation de l'entité « produit »



Par convention, l'identificateur est souligné

¹ La représentation est bien sûr simplifiée ici, dans le cadre de cette présentation.

² Dans le modèle « entité-relation », l'« identificateur » identifie de manière unique une entité ; il correspond à un ou plusieurs attributs.

DONNÉE DE LA PERSONNE

UNE PERSONNE EST SOIT UN PRODUCTEUR, SOIT UN ACHETEUR



- COMMENT REPRÉSENTER UNE PERSONNE ?

- DE QUOI A-T-ON BESOIN ?

La nom de la personne
exemple : De La Rue

Connaître son adresse dont notamment le code postal et la ville,
exemple : 80, bd. J. Jaurès - 92110 - Clichy

Connaître son numéro de téléphone
exemple : 01.47.15.30.00

- COMMENT IDENTIFIER UNE PERSONNE DE MANIÈRE UNIQUE ?

Identification avec un numéro d'ordre associé

- REPRÉSENTATION PAR LES ATTRIBUTS

1) **numéro** : nombre entier, l'« identificateur »

2) **nom** : texte

3) **adresse** : texte

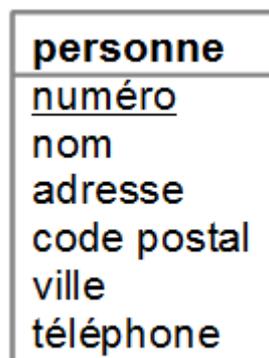
4) **code postal** : texte

5) **ville** : texte

6) **téléphone** : texte

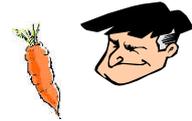
- MODÉLISATION

Représentation de l'entité « personne »



ENTITÉS ET BASE DE DONNÉES

CAS DES ENTITÉS PRODUIT ET PERSONNE



▪ TRADUCTION D'ENTITÉ DANS LA BASE DE DONNÉES

Définition d'une « table » ¹ pour une entité, avec une colonne ² par attribut et une ligne par élément de donnée (« n-uplet » ³)

L'identificateur est appelé la « clef » ; il est automatiquement créé un « index » associé dans la base pour la gestion des données ⁴

Le « schéma de la table » indique le nom de la table et la liste des attributs, avec la spécification des caractéristiques de chacun :

- indication si c'est la clef ⁵
- type de ses valeurs (« domaine »)
- donnée obligatoire ⁶ ou facultative

▪ SCHÉMA DE LA TABLE « PRODUIT »

- 1) **numero** : nombre entier (clef)
- 2) **nom** : texte obligatoire, d'au plus 40 caractères
- 3) **type** : texte obligatoire, soit LEGUME, soit FRUIT
- 4) **prix** : nombre à 2 décimales obligatoire

▪ SCHÉMA DE LA TABLE « PERSONNE »

- 1) **numero** : nombre entier (clef)
- 2) **nom** : texte obligatoire, d'au plus 40 caractères
- 3) **adresse** : texte obligatoire, d'au plus 60 caractères
- 4) **code_postal** : texte obligatoire, de 5 caractères
- 5) **ville** : texte obligatoire, d'au plus 40 caractères
- 6) **telephone** : texte facultatif, de 14 caractères

UNE BASE DE DONNÉE EST UN ENSEMBLE DE TABLES (MODÈLE RELATIONNEL)

¹ Une table est aussi appelée une « relation » (vocabulaire du modèle relationnel)

² Un attribut s'appelle aussi un « champ » ; ici, un nom d'attribut dans la table est par précaution noté sans accents et en remplaçant tout espace par un trait de souligné (« _ ») afin d'éviter des anomalies dans les traitements informatiques

³ L'élément de la table ou « n-uplet » s'appelle aussi un enregistrement, et « tuple » en anglais.

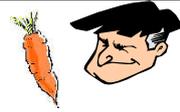
⁴ Cet index permet de retrouver rapidement un enregistrement à partir de sa clef

⁵ Si l'identificateur de l'entité est composé de plusieurs attributs, chacun de ses attributs est alors indiqué comme définissant la clef.

⁶ La valeur d'une clef est toujours obligatoire ; on peut aussi indiquer la valeur par défaut en cas d'absence de la donnée lors de l'enregistrement.

TABLES PRODUIT ET PERSONNE

EXEMPLES DE REPRÉSENTATION DANS LA BASE DE DONNÉES



▪ EXEMPLE POUR LA TABLE « PRODUIT »

<i>produit</i>				
<u>numero</u>	nom	type		prix
		LEGUME	FRUIT	
1	CAROTTE	X		1,00
2	BROCOLIS	X		2,00
3	PANAIS	X		3,00
4	KIWI		X	2,20
5	CHOU DE CHINE	X		2,50
6	POMME		X	1,60
7	POIRE		X	1,90

▪ EXEMPLE POUR LA TABLE « PERSONNE »

<i>personne</i>					
<u>numero</u>	nom	adresse	code_postal	ville	telephone
1	DE LA RUE	9, rue Convention	93100	MONTREUIL	01.48.70.60.00
2	KING KING	1, place d'Italie	75013	PARIS	01.44.08.13.13
3	LILI	3, rue Résistance	95200	SARCELLES	01.34.38.20.00
4	CESAR	80, bd. J. Jaurès	92110	CLICHY	01.47.15.30.00
5	TAO	1, avenue de Paris	78000	VERSAILLES	01.30.97.80.00
6	MOMO	6, place Gambetta	75020	PARIS	01.43.15.20.20

DONNÉE DE LA VENTE

LA VENTE D'UN PRODUIT À UNE PERSONNE



• COMMENT REPRÉSENTER UNE VENTE ?

▪ CARACTÉRISATION D'UNE VENTE

Association entre entités : un produit est vendu à une personne, à une date, avec une quantité, et à un prix éventuellement différent du prix du produit (exemple : réduction)

Identification impossible de manière unique avec ces informations :
→ identification par un numéro d'ordre

▪ REPRÉSENTATION

Association entre un produit et une personne avec les attributs :

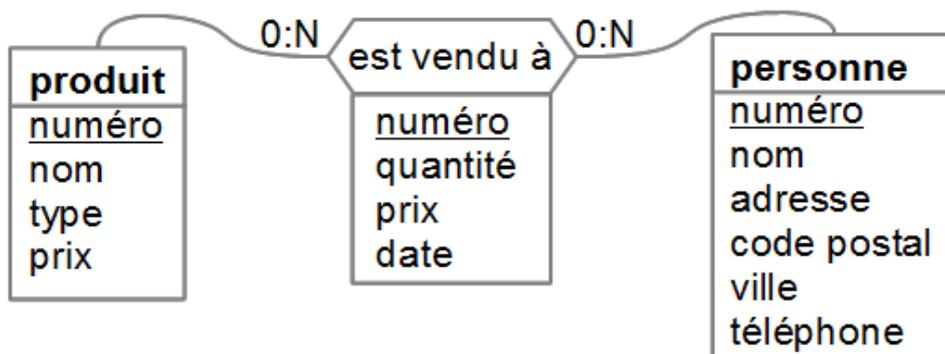
- 1) **numéro** : nombre entier, l'« identificateur »
- 2) **quantité** : nombre entier de kilogrammes achetés
- 3) **prix** : nombre entier du prix payé en euros
- 4) **date** : de la vente

▪ MODÉLISATION DE L'ASSOCIATION « VENTE »

Détermination du nombre d'associations possibles (« cardinalité ») pour chacune des entités, sous la forme « minimum : maximum »

1 produit peut ne pas être vendu ou l'être N fois ¹ → cardinalité 0 : N

1 personne peut ne pas acheter ou le faire N fois → cardinalité 0 : N



Association représentée par un polygone avec ses attributs propres, et les cardinalités notées sur les branches vers les entités

CAS D'ASSOCIATION « PLUSIEURS À PLUSIEURS »

¹ N désigne ici un nombre non limité a priori, c'est-à-dire plusieurs fois.

ASSOCIATION VENTE ET BASE DE DONNÉES

CAS DE L'ASSOCIATION DE LA VENTE



▪ TRADUCTION DANS LA BASE DE DONNÉES

Définition d'une table pour l'association, où chaque entité impliquée est représentée par sa clef ; ici, clefs d'entités associées : numéro du produit vendu et numéro de la personne acheteuse

Schéma de la table « vente » :

- 1) **numero** : nombre entier (clef)
- 2) **ref_produit** : numéro d'identification dans la table « produit »
- 3) **ref_acheteur** : numéro d'identification dans la table « personne »
- 4) **quantite** : nombre entier de kilogrammes achetés
- 5) **prix** : nombre décimal du prix payé en euros
- 6) **date** : de la vente

▪ EXEMPLE DE DONNÉES

Exemple : vente n° 1 de 30 Kg de brocolis à Lili le 27/12/2005 au prix de 1 €90 le kilogramme

<i>vente</i>					
<u>numero</u>	ref_produit	ref_acheteur	quantite	prix	date
1	2	3	30	1,90	2005-12-27
2	3	3	3	2,50	2005-12-28

<i>produit</i>				
<u>numero</u>	nom	type		prix
		LEGUME	FRUIT	
1	CAROTTE	X		1,00
2	BROCOLIS	X		2,00

<i>personne</i>					
<u>numero</u>	nom	adresse	code_postal	ville	telephone
1	DE LA RUE	9, rue Convention	93100	MONTREUIL	01.48.70.60.00
2	KING KING	1, place d'Italie	75013	PARIS	01.44.08.13.13
3	LILI	3, rue Résistance	95200	SARCELLES	01.34.38.20.00

DONNÉE DU LOT

UN LOT CORRESPOND ICI À UN SEUL PRODUIT



- COMMENT REPRÉSENTER UN LOT ?

- CARACTÉRISATION D'UN LOT

Une entité en double association avec produit et personne : un lot contient un produit, et il est fourni par une personne (producteur)

Il faut aussi connaître la taille initiale du lot, la taille du reste (pas encore vendu), la date de la fourniture et le prix d'achat (pour 1 kg)

Exemple : lot de 450 Kg de carottes entièrement vendus, fourni par De La Rue le 29 décembre 2005 au prix de 0,70 euros le kg

- COMMENT IDENTIFIER UN LOT DE MANIÈRE UNIQUE ?

Impossible de manière unique avec les informations
→ identification par un numéro d'ordre

- MODÉLISATION

Association « un produit compose un lot »

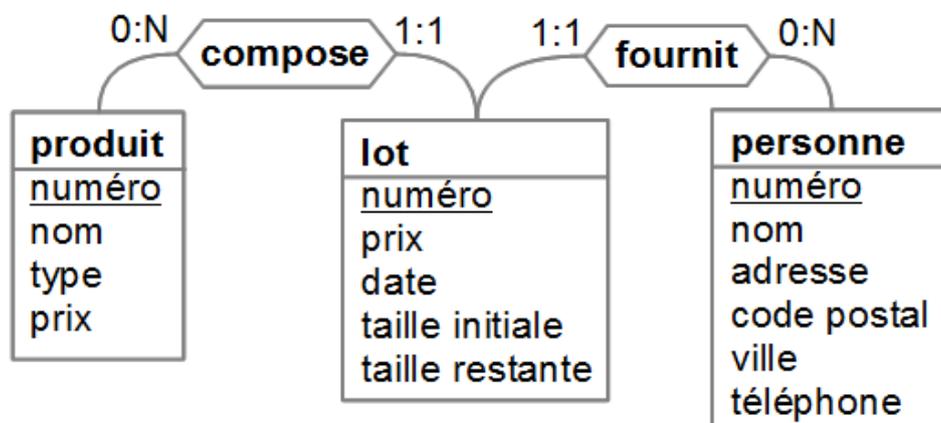
1 produit peut ne pas composer ou le faire N fois → cardinalité 0 : N

1 lot est composé d'exactly un produit → cardinalité 1 : 1

Association « un producteur fournit un lot »

1 personne peut ne pas fournir ou le faire N fois → cardinalité 0 : N

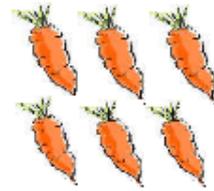
1 lot est fourni par exactly une personne → cardinalité 1 : 1



CAS PARTICULIER D'ASSOCIATIONS « 1 À PLUSIEURS »

LOT DANS LA BASE DE DONNÉES

ENTITÉ « lot » EN ASSOCIATION
AVEC « produit » ET « personne »



• REPRÉSENTATION

▪ REPRÉSENTATION DANS LA BASE DE DONNÉES

Associations « fournit » et « compose » avec une cardinalité 1:1

Schéma de la table « lot » :

- 1) **numero** : nombre entier unique d'identification (clef)
- 2) **ref_produit** : numéro d'identification dans la table « produit »
- 3) **ref_producteur** : numéro d'identification dans la table « personne »
- 4) **taille_initiale** : nombre entier de kg (taille à la livraison)
- 5) **taille_reste** : nombre entier de kg (pas encore vendus)
- 6) **prix** : nombre entier en centimes, correspondant au prix d'un kilogramme acheté au producteur
- 7) **date** : de la fourniture

▪ EXEMPLE DE DONNÉES

Le lot n° 1 : 450 Kg de carottes entièrement vendus, achetés au prix de 0,70 euros le kilogramme, le 29 décembre 2005 à De La Rue

<i>lot</i>						
<u>numero</u>	ref_produit	ref_producteur	taille_initiale	taille_reste	prix	date
1	1	1	450	0	0,70	2005-12-29
2	3	2	100	0	3,00	2005-12-29
3	1	2	300	110	0,60	2005-12-30

<i>produit</i>				
<u>numero</u>	nom	type		prix
		LEGUME	FRUIT	
1	CAROTTE	X		1,00
2	BROCOLIS	X		2,00

<i>personne</i>					
<u>numero</u>	nom	adresse	code_postal	ville	telephone
1	DE LA RUE	9, rue Convention	93100	MONTREUIL	01.48.70.60.00
2	KING KING	1, place d'Italie	75013	PARIS	01.44.08.13.13
3	LILI	3, rue Résistance	95200	SARCELLES	01.34.38.20.00

ASSOCIATION ET BASE DE DONNÉES

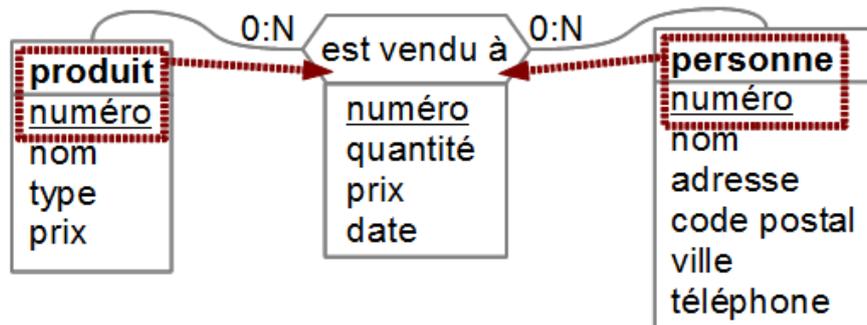
TRADUCTION DANS LE MODÈLE RELATIONNEL



■ CAS GÉNÉRAL D'UNE ASSOCIATION SANS CARDINALITÉ 1:1 ¹

Association représentée par une table, où chaque entité impliquée dans l'association y est représentée par sa clef en tant qu'attribut

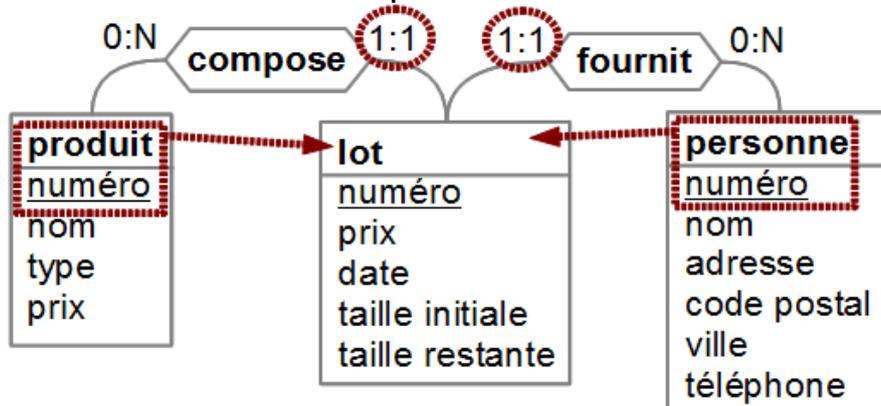
Exemple : association « est vendu à »



■ CAS PARTICULIER D'ASSOCIATION AVEC UNE CARDINALITÉ 1:1

Association représentée dans l'entité à cardinalité 1:1, par l'identificateur de l'autre entité liée en tant qu'attribut

Exemple : associations « compose » et « fournit »



■ CLEF PRIMAIRE ET CLEF ÉTRANGÈRE

La clef représentant une entité dans la table d'une association est appelée « **clef étrangère** » par opposition à la « **clef primaire** » servant d'identificateur dans la table de l'entité ²

Exemple : `ref_acheteur` est une clef étrangère dans la table `vente`, `numéro` est la clef primaire dans la table `personne`

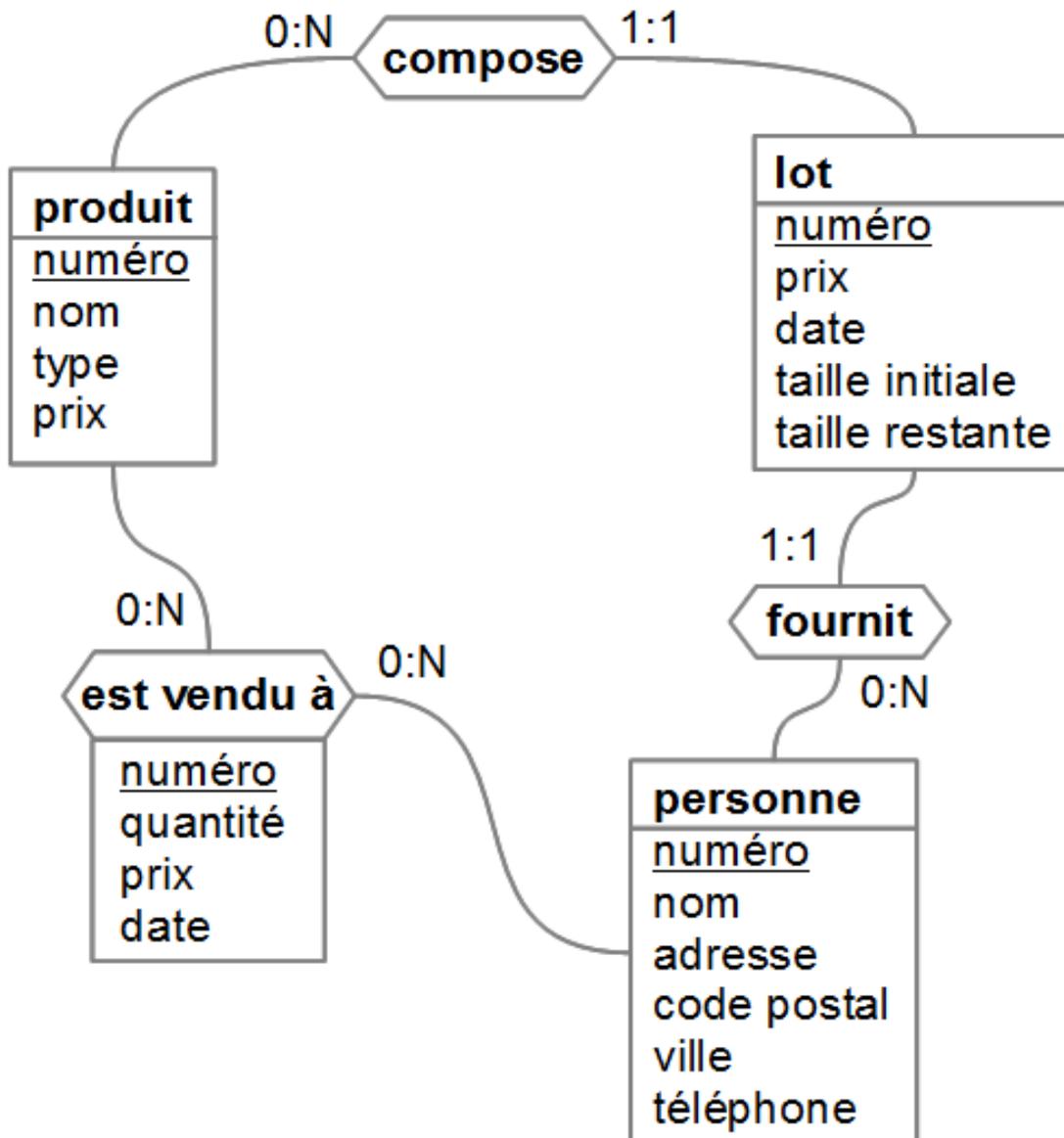
UNE ASSOCIATION EST REPRÉSENTÉE EN GÉNÉRAL PAR UNE TABLE, SAUF DANS LE CAS OU UNE DES BRANCHES PORTE UNE CARDINALITÉ 1:1

¹ En fait, il s'agit d'une cardinalité maximale de 1 ; ainsi, cela peut aussi être 0:1

² Une clef étrangère est aussi appelée « clef externe » et une clef primaire « clef de relation »

MODÉLISATION DE LA COOPÉRATIVE

MODÈLE DE DONNÉES GLOBAL



POUR UNE RÉALITÉ DONNÉE, PLUSIEURS MODÈLES SONT POSSIBLES

EXERCICES D'EXTENSION DU MODÈLE

1) Un lot peut être vérifié une ou plusieurs fois par un employé (éventuellement différent à chaque fois) afin de déterminer son état général (bon, dégradé ou perdu) ; un employé est caractérisé par sa fonction et son poste téléphonique.

Exemple : le lot n° 1, vérifié le 30/12/2005 comme bon par Joyeux puis dégradé le 3/1/2006 par Lapin.

Proposer une modélisation de l'employé et puis de la vérification

2) La coopérative offre des promotions sous la forme d'un panier de 2 produits, avec un prix attractif au kilogramme

Exemple : le panier « purée hivernale », à 3,2 euros, composé de 1 Kg de carottes et de 1 Kg de panais

a) Proposer une modélisation de ce panier

b) Quelles sont les conséquences sur les autres entités ?

REDONDANCE DES DONNÉES

DUPLICATION INUTILE DE DONNÉES



• CAS DE REDONDANCE

▪ EXEMPLE

On définit un référent unique pour toute personne enregistrée, qui correspond à un employé de la coopérative en contact avec elle, identifié par le nom de cet employé et son poste téléphonique

Si on rajoute ces attributs dans la table « personne » :

<i>personne</i>					
<u>numero</u>	nom	...	telephone	referent	poste
1	DE LA RUE	...	01.48.70.60.00	GERARD	223
2	KING KING	...	01.44.08.13.13	GARANCE	222
3	LILI	...	01.34.38.20.00	GARANCE	222

on voit apparaître rapidement une duplication d'information inutile et dangereuse car si par exemple le poste téléphonique d'un référent change, il faut alors répercuter la modification à chaque répétition

▪ CONSÉQUENCES

Gâchis de mémoire en cas de répétitions nombreuses

Risques élevés d'incohérence lors de mises à jour incomplètes

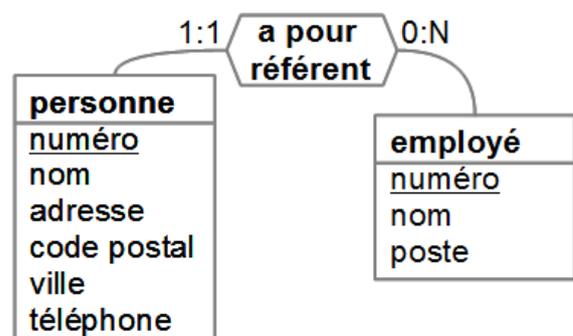
▪ ANALYSE

Cas général de « dépendance fonctionnelle » entre 2 attributs d'un enregistrement : la connaissance de la valeur du premier attribut entraîne la détermination du second attribut de manière unique

Exemple : si on connaît le nom du référent, on peut trouver le poste

▪ SOLUTION

Elimination des dépendances fonctionnelles par décomposition de l'entité en plusieurs entités ; exemple :



LA REDONDANCE EST UN DÉFAUT DANS UNE BASE DE DONNÉES

INCOHÉRENCE DES DONNÉES

CAS OÙ LES DONNÉES ENREGISTRÉES SONT INCORRECTES



● CAS D'INCOHÉRENCE

- EXEMPLE DE RÉFÉRENCE INCORRECTE : INCOHÉRENCE RÉFÉRENTIELLE

Cas où un lot fait référence à un produit non enregistré

<i>lot</i>						
<u>numero</u>	ref_produit	ref_producteur	taille_initiale	taille_reste	prix	date
1	1	1	450	0	0,70	2005-12-29
2	3	2	100	0	3,00	2005-12-29
3	1	2	300	110	0,60	2005-12-30

<i>produit</i>				
<u>numero</u>	nom	type		prix
		LEGUME	FRUIT	
1	CAROTTE	X		1,00
2	BROCOLIS	X		2,00
4	KIWI		X	2,20

Cause possible de cette situation incohérente

suppression brutale de l'enregistrement n° 3 dans la table *produit*

- AUTRES CAS : INCOHÉRENCE DE VALEUR

Valeur obligatoire absente, ou en dehors du domaine de la colonne

● CONTRAINTES DE COHÉRENCE

- CONTRAINTE DE CLEF

Toute valeur d'une clef est unique

- CONTRAINTE DE DOMAINE

Toute valeur d'une colonne doit correspondre à son domaine

Exemple : la valeur d'un type de produit est soit LEGUME, soit FRUIT

- CONTRAINTE D'INTÉGRITÉ RÉFÉRENTIELLE

Toute valeur d'une clef étrangère doit exister pour la clef primaire

Exemple : toute référence à un produit doit exister dans la table *produit*

LA COHÉRENCE EST UNE QUALITÉ INDISPENSABLE DES BASES DE DONNÉES

EXERCICE SUR LA COHÉRENCE

DANS LE CADRE DU SYSTÈME D'INFORMATION DE LA COOPÉRATIVE,

INDIQUER LES CONTRAINTES D'INTÉGRITÉ RELATIONNELLE

EXISTE-T-IL D'AUTRES CONTRAINTES DE COHÉRENCE ? LESQUELLES ?

MODÉLISATION CONCEPTUELLE : RÉCAPITULATIF

PROCÉDURE D'ANALYSE ET DE REPRÉSENTATION DES DONNÉES



- DÉCOMPOSITION EN ENTITÉS

- INVENTAIRE DES ÉLÉMENTS DE LA RÉALITÉ
identification d'éléments indépendants
- DÉTERMINATION DES ATTRIBUTS POUR CHAQUE ENTITÉ
liste des données atomiques, avec leurs formats de valeurs
- CHOIX DE L'IDENTIFICATEUR
un ou plusieurs des attributs

- IDENTIFICATION DES ASSOCIATIONS

- ASSOCIATIONS EXISTANTES
associations entre deux ou plus entités
- DÉTERMINATION DES CARACTÉRISTIQUES DE CHAQUE ASSOCIATION
liste des éventuels attributs spécifiques et des cardinalités

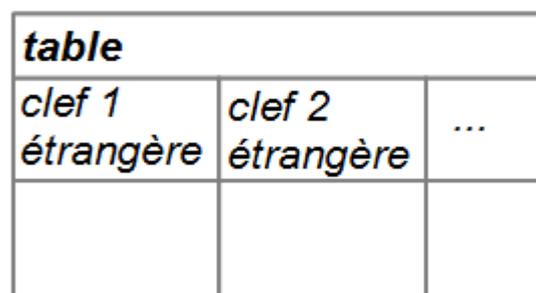
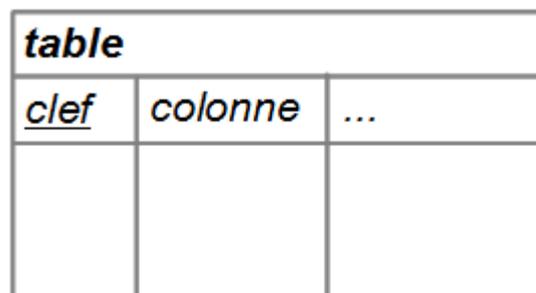
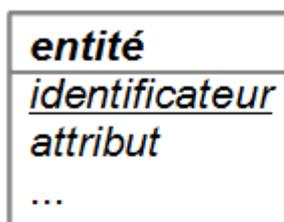
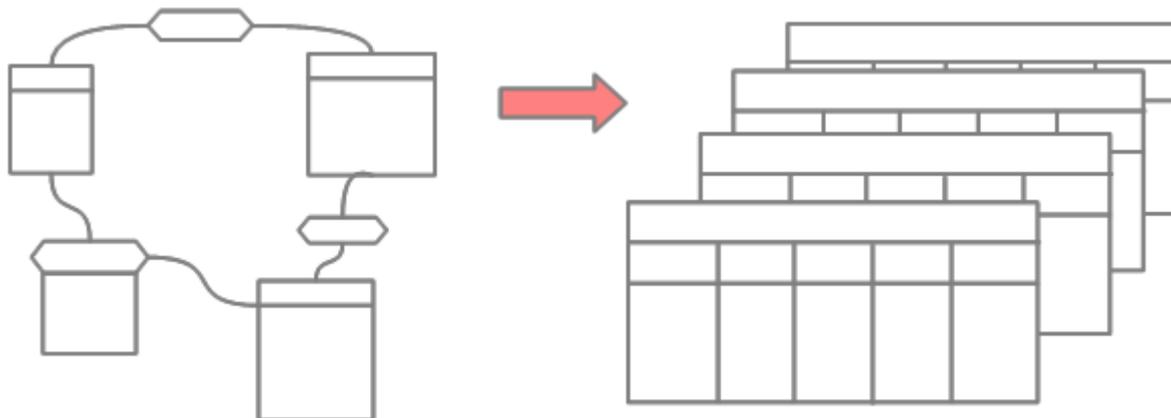
- VALIDATION DU MODÈLE CONCEPTUEL

- ELIMINATION DES REDONDANCES
redécomposition d'entité si besoin
- DÉTERMINATION DES CONTRAINTES
valeurs obligatoires, intégrité référentielle
- CONFRONTATION AUX BESOINS
vérification de la possibilité de répondre à toutes les questions

*ÉTAPE DE MODÉLISATION CONCEPTUELLE CRUCIALE,
SITUÉE ENTRE L'ANALYSE DES BESOINS ET
LA CRÉATION DE LA BASE DE DONNÉES*

PASSAGE À LA BASE DE DONNÉES

TRADUCTION DES ENTITÉS-ASSOCIATIONS (MODÈLE CONCEPTUEL)
EN UNE BASE DE DONNÉES (MODÈLE RELATIONNEL)



clef étrangère

OPÉRATION DE TRADUCTION AUTOMATISABLE

EXERCICES SUR LA MODÉLISATION

1. Comment définit-on une entité ?
2. Quelles sont les caractéristiques d'une association ?
3. Quel est le rôle d'un identificateur ?
4. Quels sont les principaux types de données ?
5. Que signifie une cardinalité de type « 1:1 » et « 0:1 » ?
6. A quoi correspondent une entité et un identificateur dans le modèle relationnel ?
7. Quelle est la différence entre une clef primaire et une clef étrangère ?
8. Citer les types de contraintes de cohérence
9. Qu'est-ce qu'une dépendance fonctionnelle ?
10. Comment interpréter une association du type « 1 à 1 » ?

UTILISATION DES BASES DE DONNÉES

Langage SQL

Outils EasyPHP et phpMyAdmin

Définition de la structure d'une base

Langage de manipulation des données

INTRODUCTION

UTILISATION DES BASES DE DONNÉES ¹

• SQL ², UN LANGAGE POUR LES BASES DE DONNÉES

▪ LANGAGE SIMPLE ET NORMALISÉ

Destiné à faciliter les manipulations de données

Fondé sur le modèle relationnel de bases de données

Ecriture d'une commande sous la forme d'une « requête »

▪ PRINCIPALES FONCTIONS

a) Définition de la structure d'une base (schémas de tables, etc.)

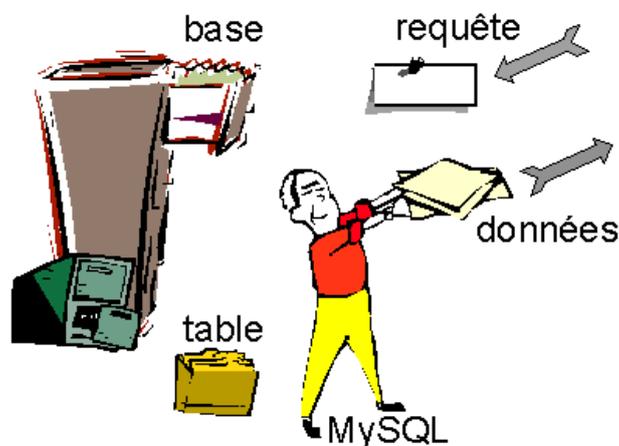
b) Manipulation des données (recherche, ajout, mise à jour, etc.)

• OUTILS ASSOCIÉS

▪ SYSTÈME DE GESTION DE BASE DE DONNÉES (SGBD)

Ensemble de logiciels gérant des bases de données,

exemple : MySQL (SGBD libre très utilisé sur le Web) ³



▪ PROGRAMMATION

Interfaçage avec les principaux langages : Java, PHP, C++, etc.

▪ UTILISATION SUR LE WEB

Serveur Web + SGDB + interfaces de programmation + ...

Exemple : Apache + MySQL + PHP + phpMyAdmin

→ EasyPHP (Windows) ou MAMP (Mac OS X)

¹ En anglais, une base de donnée se dit *database*.

² SQL : *structured query language*, un langage de requêtes structurées ; présentation ici des bases de la version 2 de 1992 ou SQL-92.

³ Autres SGBD répandus : Oracle, PostgreSQL, Access, etc.

CAS DE EASYPHP SOUS WINDOWS

APERÇU DE LA VERSION 5.3 D'EASYPHP
(<http://www.easyphp.org/> ¹)



• CARACTÉRISTIQUES GÉNÉRALES

- UN ENVIRONNEMENT COMPLET SOUS WINDOWS
libre de droits pour un usage non commercial
serveur Apache + moteur PHP + SGBD MySQL + PhpMyAdmin
- ADMINISTRATION SIMPLIFIÉE
manipulations facilitées via un navigateur
outil PhpMyAdmin pour l'administration des bases de données

• INSTALLATION

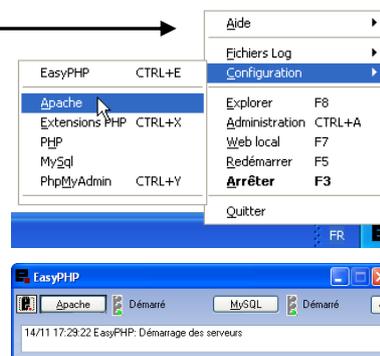
- APPLICATION EASY-PHP
environnement complet en C:\Program Files\Easy-PHP...
lancement par DÉMARRER PROGRAMMES ... EASYPHP ²
icône à droite dans la barre de tâches
vérification du fonctionnement par consultation du site local
adresse : <http://localhost/> ou <http://127.0.0.1/> ³

- UTILISATION DE L'ICÔNE EN BAS À DROITE

clic-droit : menu de commande

- page d'administration : ADMINISTRATION
- accès au dossier des pages : EXPLORER
- redémarrage des serveurs : REDÉMARRER
- arrêt des serveurs : ARRÊTER

double-clic sur l'icône : état courant



ATTENTION ! PRÉSENTATION RAPIDE D'EASYPHP
POUR UN USAGE PERSONNEL, INTERMITTENT ET NON PERMANENT
AUTRE SERVEUR GRATUIT POUR MAC OS X : MAMP ⁴

¹ A AgroParisTech, dans la boîte à outils de Gaia : <https://gaia.agroparistech.fr/>

² En salle d'informatique à AgroParisTech, commande dans le sous-menu INFO et puis au message VOUS N'ÊTES PAS ADMINISTRATEUR... CONTINUER ? confirmer par OUI. Sous Windows Vista et 7, il faut lancer l'application en mode administrateur : clic-droit sur l'icône de raccourci vers le programme et EXÉCUTER EN TANT QU'ADMINISTRATEUR.

³ localhost et 127.0.0.1 : adresses pour désigner le poste courant dans l'Internet

⁴ MAMP disponible en : <http://www.mamp.info/>

RÉGLAGES DE EASYPHP

DOSSIERS POUR LES PAGES WEB ET LES BASES DE DONNÉES ¹



● APACHE

- ACCÈS AU FICHIER DE CONFIGURATION `httpd.conf` ²
Clic-doit sur l'icône d'EasyPHP puis CONFIGURATION APACHE
- RÉGLAGE DU DOSSIER SPÉCIFIQUE AUX PAGES WEB
Création d'un dossier ³, exemple : `D:\easyphp\www`
Remplacements dans le texte du fichier de configuration
 - de : `DocumentRoot "${path}/www"`
par : `DocumentRoot "D:/easyphp/www"`
 - de : `<Directory "${path}/www">`
par : `<Directory "D:/easyphp/www">`

● MYSQL

- ACCÈS AU FICHIER DE CONFIGURATION `my.ini` ⁴
Clic-doit sur l'icône d'EasyPHP puis CONFIGURATION MYSQL
- RÉGLAGE DU DOSSIER SPÉCIFIQUE AUX BASES DE DONNÉES
Création d'un dossier ³, exemple : `D:\easyphp\data`
Remplacements dans le texte du fichier de configuration
 - de : `datadir=${path}/mysql/data/`
 - par : `datadir=D:/easyphp/data/`

Attention ! sous-dossier `mysql` à copier
dans le dossier `D:\easyphp\data`
à partir du dossier : `C:\Program Files\Easy-PHP...\mysql\data`

Ne pas oublier de relancer le serveur après toute modification :
clic-doit sur l'icône d'EasyPHP puis REDÉMARRER

RECONFIGURATIONS NON INDISPENSABLES MAIS CONSEILLÉES ⁵

¹ Dans les salles d'informatique d'AgroParisTech : c'est déjà réglé en `N:\www` et `N:\data`

² Ce fichier `httpd.conf` est en fait dans le dossier `C:\Program Files\Easy-PHP...\conf_files\`

³ A priori, en dehors du dossier des programmes sous Windows (`C:\Program Files`)

⁴ Ce fichier `my.ini` est en fait dans le dossier `C:\Program Files\Easy-PHP...\conf_files\`

⁵ Afin de limiter la perte de données en cas de modification du logiciel et de faciliter leurs sauvegardes

OUTIL PHPMYADMIN

ADMINISTRATION DES BASES DE DONNÉES AVEC MYSQL



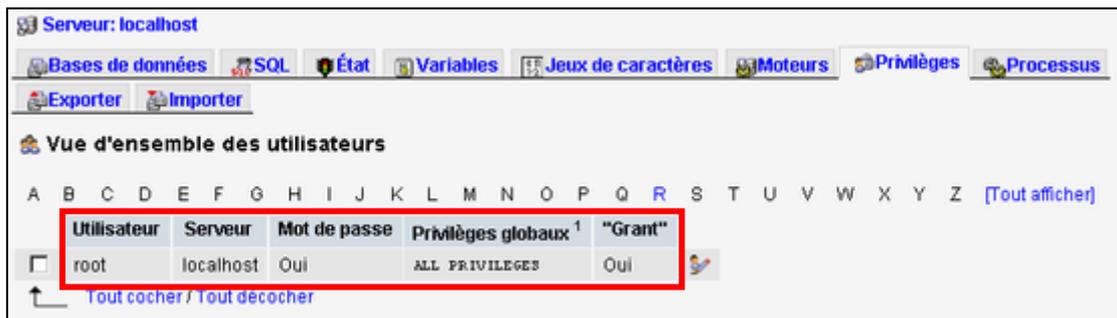
■ LANCEMENT DANS EASYPHP

clic-droit sur l'icône d'EasyPHP puis ADMINISTRATION et puis au milieu de la page, ADMINISTRER MYSQL AVEC PHPMYADMIN



■ VÉRIFICATION DES DROITS D'UTILISATION (« PRIVILÈGES »)

Onglet PRIVILÈGES dans la page de phpMyAdmin



Ici l'utilisateur `root` dispose de tous les droits sur les bases, et a besoin d'un mot de passe pour s'identifier ¹

PHPMYADMIN : MANIPULATIONS INTERACTIVES DES BASES DE DONNÉE

¹ L'absence de mot de passe est une situation à éviter sur un poste à usage strictement personnel, et c'est inacceptable sur un vrai serveur (possibilité de piratage) ; dans Easy-PHP, le mot de passe est connu par PHPMyAdmin et on est automatiquement l'utilisateur `root` (terme du jargon des serveurs qui désigne l'administrateur du serveur disposant de tous les droits)

CRÉATION D'UNE BASE

CAS DE LA BASE POUR LA COOPÉRATIVE

• CRÉATION DE LA BASE AVEC PHPMYADMIN

- FORMULAIRE DANS LA PAGE D'ACCUEIL (RETOUR VIA LE BOUTON )

Indiquer le nom ¹ de la base à CRÉER UNE BASE DE DONNÉES



puis à INTERCLASSEMENT choisir UTF8_GENERAL_CI ²
et valider avec le bouton CRÉER



• CRÉATION DE LA BASE EN SQL

- COMMANDE « CREATE DATABASE »

Syntaxe de la requête ³ : `create database nom`

Exemple : `create database coop` ⁴

CRÉATION D'UNE NOUVELLE BASE, VIDE

¹ Un nom de base est a priori constitué d'un maximum de 64 lettres, chiffres ou trait de souligné (« _ ») tout en commençant par une lettre ; il est conseillé d'éviter les lettres accentuées

² L'interclassement définit la méthode de comparaison et de tri des symboles pour un alphabet donné, en considérant les variantes accentuées ou de casse ; exemple : égalité entre « u », « U », « ù » et « Û »

³ Par convention les noms de commandes définis dans le langage SQL sont écrits en lettres majuscules mais en fait les SGBD ne font pas la différence entre les majuscules et les minuscules : on peut donc taper la requête avec les noms de commande écrits en minuscules

⁴ Dans les affichages du résultat d'une requête MySQL, les noms de table apparaissent entourés par un accent grave (« ` ») : ne pas tenir compte de cette spécificité de MySQL

CRÉATION D'UNE TABLE AVEC PHPMYADMIN

CAS DE LA TABLE « produit » DE LA BASE « coop »

■ FORMULAIRE DANS LA PAGE DE LA BASE

Indiquer le nom ¹ de la table à CRÉER UNE NOUVELLE TABLE... et le nombre de champs de la table, puis valider avec le bouton EXÉCUTER

Créer une nouvelle table sur la base **coop**

Nom: Nombre de champs:



Champ	Type (T)	Taille/Valeurs ¹	Défaut ²	Interclassement	Attributs	Null	Index	A_I
	INT		aucune			<input type="checkbox"/>	--	<input type="checkbox"/>
	INT		aucune			<input type="checkbox"/>	--	<input type="checkbox"/>
	INT		aucune			<input type="checkbox"/>	--	<input type="checkbox"/>
	INT		aucune			<input type="checkbox"/>	--	<input type="checkbox"/>

■ FORMULAIRE DE DÉFINITION DES COLONNES DE LA TABLE

Définition de chaque colonne de la table à partir de la modélisation

Colonne	Type (T)	Taille/Valeurs ¹	Défaut ²	Interclassement	Attributs	Null	Index	A_I
numero	INT		Aucun		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
nom	VARCHAR	40	Aucun			<input type="checkbox"/>	--	<input type="checkbox"/>
type	ENUM	'LE', 'FRUIT'	Aucun			<input type="checkbox"/>	--	<input type="checkbox"/>
prix	DECIMAL	7,2	Tel que défini : 0			<input type="checkbox"/>	--	<input type="checkbox"/>

selon les indications suivantes :

- numero : type « int », attribut « unsigned », index « primary » car clef, et cocher « A_I » (*auto_increment*) pour affectation automatique du n° d'ordre
 - nom : type « varchar », taille « 40 » et on considère la valeur obligatoire ²
 - type : type « enum », valeurs « 'LEGUME', 'FRUIT' », valeur obligatoire
 - prix : type « decimal », attribut « 7,2 », défaut « Tel que défini » et « 0 » afin de placer automatiquement la valeur zéro si le prix est inconnu
- puis après vérifications, terminer en cliquant sur SAUVEGARDER



	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/>	<u>numero</u>	int(10)		UNSIGNED	Non	Aucun	AUTO_INCREMENT
<input type="checkbox"/>	nom	varchar(40)	utf8_general_ci		Non	Aucun	
<input type="checkbox"/>	type	enum('LEGUME','FRUIT')	utf8_general_ci		Non	Aucun	
<input type="checkbox"/>	prix	decimal(7,2)			Non	0.00	

¹ Un nom de table est a priori constitué d'un maximum de 64 lettres, chiffres ou trait de souligné (« _ ») tout en commençant par une lettre ; il est conseillé d'éviter les lettres accentuées

² Dans le cas de valeur obligatoire (case Null non cochée), le champ doit comporter une valeur et le SGBD refusera un ajout si la valeur est absente ; attention ! la chaîne vide (« ' ' ») constitue une valeur.

CRÉATION D'UNE TABLE EN SQL

▪ DÉFINITION D'UN CHAMP

Un champ est défini par : nom ¹, type et compléments éventuels

Principaux types de champ

- nombre entier : `integer` avec signe, ou `integer unsigned` sans ²
synonyme possible de `integer` : `int`
possibilité d'un nombre entier réduit : `smallint` ³
- nombre décimal : `decimal(précision, décimales)`
avec indication du nombre maximal de chiffres (*précision*) dont ceux après la virgule décimale (*décimales*)
synonymes possibles de `decimal` : `dec` ou `numeric`
cas de calculs financiers avec une précision au chiffre près
exemple : `decimal(7,2)` pour une valeur entre -99999,99 et 99999,99
- nombre réel : `float` ou `double precision` (précision maximale)
cas de calculs scientifiques sans nécessité d'exactitude ⁴
- texte : `varchar(taille)` avec au-maximum *taille* caractères
- valeur parmi une liste de possibilités : `enum(liste)`
- date : `date`

Principaux compléments

- incrémentation automatique (MySQL) : `auto_increment`
le SGBD MySQL attribue automatiquement un numéro d'ordre au champ si la valeur indiquée est vide ou nulle
- indication de valeur obligatoire : `not null`
`null` désigne l'absence de valeur et non pas une valeur nulle (0)
- valeur choisie en absence de définition : `default valeur`

▪ DÉFINITION DE LA CLEF

Indication du champ de la clef ⁵ : `primary key (champ)`

ou s'il s'agit de plusieurs champs : `primary key (liste de champs)`

¹ Un nom de champ est a priori constitué d'un maximum de 64 lettres, chiffres ou trait de souligné (« _ ») tout en commençant par une lettre ; il est conseillé d'éviter les lettres accentuées

² MySQL offre la possibilité d'indiquer un nombre maximal de chiffres lors de la définition mais cela n'est pas dans la norme de SQL et ne concerne que l'affichage et non le stockage de la valeur.

³ A priori, la plage de valeurs d'un nombre entier ordinaire s'étend de - 2 147 483 648 à 2 147 483 647 ou de 0 à 4 294 967 295 si sans signe (codage sur 4 octets) ; dans le cas d'un entier réduit, l'intervalle est de - 32 768 à 32 767 ou de 0 à 65 535 (codage sur 2 octets).

⁴ Par rapport au type `DECIMAL`, le type `FLOAT` ne garantit pas l'exactitude des calculs au chiffre près mais ceux-ci sont effectués plus rapidement.

⁵ Si la clef est sur un seul champ, MySQL autorise son indication en complément de la définition du champ par la notation `primary key`

CRÉATION D'UNE TABLE EN SQL (SUITE)

▪ PRINCIPALES NOTATIONS DE VALEURS

- nombre entier : 1235 -69
- nombre réel : 3.14159 -1.8E3
- texte noté par une chaîne de caractères entre apostrophes ¹ :
'Carotte' 'aujourd'hui' '' (cas du texte vide)
- date (année-mois-jour ²) : '2009-01-07' ou '2009-1-7'
- liste (valeurs séparées par une virgule) : 'LEGUME', 'FRUIT'
avec la possibilité d'ajouter des espaces autour de la virgule

▪ COMMANDE DE CRÉATION EN SQL : « CREATE TABLE »

Syntaxe de la requête

`create table nom (liste des définitions de champ et de la clef)`
possibilité d'ajouter des espaces ou des retours à la ligne dans la notation afin de l'aérer et de faciliter la lecture ³

Exemple (MySQL)

```
create table produit (  
  numero integer unsigned not null auto_increment,  
  nom varchar(40) not null,  
  type enum('legume','fruit') not null,  
  prix integer unsigned not null default '0',  
  primary key (numero)  
)
```

▪ COMMANDE D'INFORMATION SUR UNE TABLE EN SQL : « SHOW FIELDS »

Syntaxe de la requête

`show fields from nom`

Exemple (MySQL)

`show fields from produit`

Field	Type	Null	Key	Default	Extra
numero	int(10) unsigned	NO	PRI	NULL	auto_increment
nom	varchar(40)	NO		NULL	
type	enum('LEGUME','FRUIT')	NO		NULL	
prix	decimal(7,2)	NO		0.00	

¹ Si le texte comporte une apostrophe, il faut alors la doubler ; la chaîne vide (« '' ») est une valeur.

² Notation internationale d'une date (« année-mois-jour ») permettant un classement des dates par la comparaison un à un des symboles du texte de leurs notations.

³ Attention ! pas de retour à la ligne à l'intérieur des apostrophes (« ' ») d'une chaîne de caractères .

INTÉGRITÉ RÉFÉRENTIELLE EN MYSQL

MANIPULATIONS SPÉCIFIQUES À MYSQL (VERSION 5.1)

▪ PRINCIPES DE LA MISE EN OEUVRE DE L'INTÉGRITÉ RÉFÉRENTIELLE

Procédure en 3 étapes

- 1) Choix d'un moteur de stockage ¹ particulier pour chaque table
- 2) Définition d'un index pour chaque clef étrangère
- 3) Mise en relation de la clef étrangère avec la clef primaire associée

Mécanisme de contrôle automatique

vérification lors de l'insertion de la validité de la clef étrangère

▪ MODIFICATION DU MOTEUR DE STOCKAGE D'UNE TABLE

Sélectionner d'abord la table dans le panneau de gauche

Passer dans l'onglet OPÉRATIONS

Régler MOTEUR DE STOCKAGE à INNODB

Options pour cette table

Changer le nom de la table pour

Commentaires sur la table

Moteur de stockage ⓘ **InnoDB** ▼

Interclassement ▼

PACK_KEYS ▼

CHECKSUM

DELAY_KEY_WRITE

AUTO_INCREMENT

ROW_FORMAT ▼

Exécuter

Valider via le bouton EXÉCUTER

¹ Moteur de stockage de MySQL : mode de représentation des tables offrant ou non certaines possibilités telles l'utilisation de contrainte d'intégrité référentielle (cas de InnoDB mais pas de MyISAM)

INTÉGRITÉ RÉFÉRENTIELLE EN MYSQL (SUITE)

MANIPULATIONS SPÉCIFIQUES À MYSQL (VERSION 5.1)

■ DÉFINITION D'UN INDEX POUR UNE CLEF ÉTRANGÈRE

Sélectionner d'abord la table dans le panneau de gauche

Passer dans l'onglet STRUCTURE

Dans la ligne de la clef étrangère, cliquer sur le bouton à l'éclair : 

	Colonne	Type	Interclassement	Attributs	Null	Défaut	Extra	Action					
<input type="checkbox"/>	numero	int(10)		UNSIGNED	Non	Aucun	AUTO_INCREMENT						
<input type="checkbox"/>	ref_produit	int(10)		UNSIGNED	Non	0							
<input type="checkbox"/>	ref_acheteur	int(10)		UNSIGNED	Non	0							
<input type="checkbox"/>	quantite	int(10)		UNSIGNED	Non	0							
<input type="checkbox"/>	prix	int(10)		UNSIGNED	Non	0							
<input type="checkbox"/>	date	date			Non	0000-00-00							

Il doit ensuite apparaître l'index de même nom que la clef étrangère

Index: ?

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
		PRIMARY	BTREE	Oui	Non	numero	33	A	
		ref_produit	BTREE	Non	Non	ref_produit	8	A	

■ MISE EN RELATION DE LA CLEF ÉTRANGÈRE AVEC SA CLEF PRIMAIRE

Sélectionner d'abord la table dans le panneau de gauche

Passer dans l'onglet STRUCTURE

Sous la liste des colonnes, cliquer sur le lien  [Gestion des relations](#)

Choisir ensuite la clef primaire associée à chaque clef étrangère (attention ! ne pas utiliser la première zone, « RELATIONS INTERNES »)

Relié à

	Relations internes ¹	FOREIGN KEY (INNODB)
numero		ON DELETE <input type="text"/> ON UPDATE <input type="text"/>
ref_produit	<input type="text"/>	<input type="text"/> ON DELETE <input type="text"/> ON UPDATE <input type="text"/>
ref_acheteur	<input type="text"/>	<input type="text"/> ON DELETE <input type="text"/> ON UPDATE <input type="text"/>
quantite	<input type="text"/>	Aucun index n'est défini!
prix	<input type="text"/>	Aucun index n'est défini!
date	<input type="text"/>	Aucun index n'est défini!

Valider via le bouton SAUVEGARDER

Si message d'erreur « CANNOT ADD OR UPDATE A CHILD ROW: A FOREIGN KEY CONSTRAINT FAILS », vérifier les valeurs de la clef étrangère

INTÉGRITÉ RÉFÉRENTIELLE EN MYSQL (FIN)

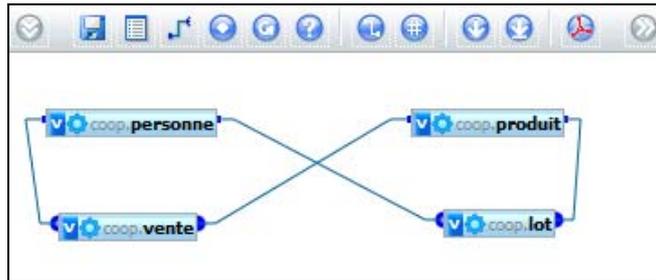
VISUALISATION DES RÉFÉRENCES ENTRE LES TABLES

■ OUTIL « CONCEPTEUR »

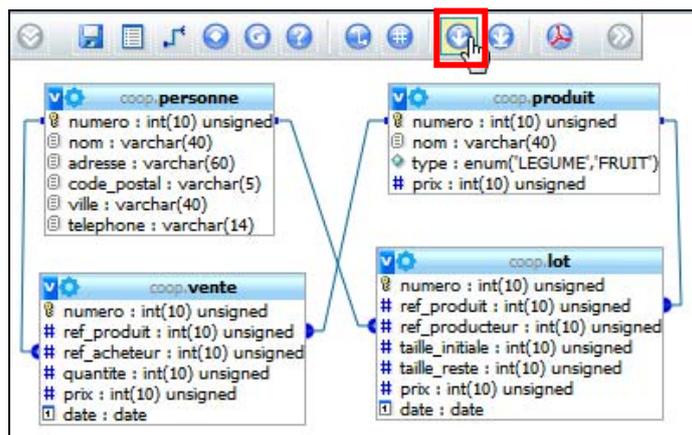
Sélectionner d'abord la base dans le panneau de gauche

Passer dans l'onglet CONCEPTEUR

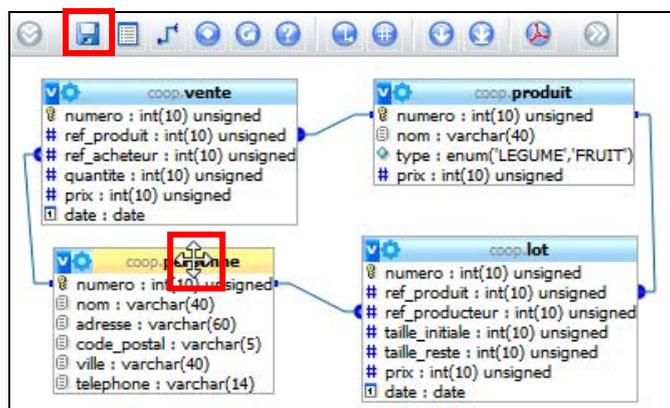
Il doit s'afficher un schéma avec les tables et leurs références



Si besoin, cliquer sur le bouton  pour visualiser les colonnes



Si besoin, faire glisser chaque table pour améliorer la visualisation



Après réorganisation, cliquer sur le bouton  pour la sauvegarder

Pour quitter le concepteur, cliquer à gauche sur le nom de la base

AJOUT DE DONNÉES

INSERTION D'UN ENREGISTREMENT DANS UNE TABLE DE LA BASE

• INSERTION DANS UNE TABLE AVEC PHPMYADMIN

Utiliser l'onglet INSÉRER

Server: localhost ▶ Base de données: coop ▶ Table: produit

Structure Afficher SQL Rechercher **Insérer** Exporter Opérations Vider Supprimer

Colonne	Type	Fonction	Null	Valeur
numero	int(10) unsigned			
nom	varchar(40)			CAROTTE
type	enum	--		LEGUME
prix	decimal(7,2)			1.00

Exécuter

Ignorer

Colonne	Type	Fonction	Null	Valeur
numero	int(10) unsigned			
nom	varchar(40)			
type	enum	--		
prix	decimal(7,2)			0.00

Exécuter

saisir les valeurs d'un enregistrement sauf pour la clef (vide ¹)
puis valider avec le bouton EXÉCUTER

• INSERTION DANS UNE TABLE AVEC SQL

▪ COMMANDE « INSERT »

Syntaxe de la requête

```
insert into table (liste de champs) values (liste de valeurs)
```

Exemple

```
insert into produit (numero, nom, type, prix )  
values (null, 'CAROTTE', 'LEGUME', 1)
```

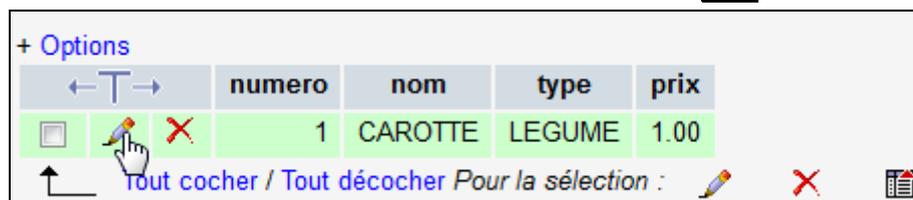
¹ Convention dans le cas d'une incrémentation automatique (*auto-increment*) par MySQL ; le vide s'indique aussi par le mot `null`.

MISE À JOUR DE DONNÉES

MODIFICATION D'UN ENREGISTREMENT DANS UNE TABLE DE LA BASE

• MISE À JOUR DANS UNE TABLE AVEC PHPMYADMIN

Afficher les enregistrements et utiliser le bouton 



Colonne	Type	Fonction	Null	Valeur
numero	int(10) unsigned			1
nom	varchar(40)			CAROTTE
type	enum	--		LEGUME
prix	decimal(7,2)			1.20

Modifier dans le formulaire puis valider avec le bouton EXÉCUTER

• MISE À JOUR DANS UNE TABLE AVEC SQL

▪ COMMANDE « UPDATE »

Syntaxes de la requête (versions simplifiées ¹)

```
update table set champ1 = valeur1 where clef = valeur
```

```
update table set champ1 = valeur1 (mise à jour de tous les enregistrements)
```

```
update table set champ1 = valeur1, champ2 = valeur2, ... where clef = valeur
```

Exemples

```
update produit set prix = 1.2 where numero = 1
```

```
update produit set nom = 'KIWI', type = 'FRUIT', prix = 1.2  
where numero = 1
```

¹ La condition suivant le mot-clef `where` peut en fait être plus développée (cf. page 42) et la modification peut alors s'appliquer à plusieurs enregistrements (et non plus un seul).

SUPPRESSION DE DONNÉES

SUPPRESSION D'UN ENREGISTREMENT DANS UNE TABLE DE LA BASE

● SUPPRESSION DANS UNE TABLE AVEC PHPMYADMIN

Afficher les enregistrements et utiliser le bouton 



	numero	nom	type	prix
<input type="checkbox"/>  	1	CAROTTE	LEGUME	1.00
<input type="checkbox"/>  	2	POTIMARRON	LEGUME	2.80

Tout cocher / Tout décocher Pour la sélection :   



Voulez-vous vraiment effectuer :
DELETE FROM `coop`.`produit` WHERE `produit`.`numero` = 2

puis confirmer la demande de suppression

● SUPPRESSION DANS UNE TABLE AVEC SQL

▪ COMMANDE « DELETE »

Syntaxe de la requête (version simplifiée ¹)

```
delete from table where clef = valeur
```

Exemples

```
delete from produit where numero = 2
```

La clef d'un enregistrement supprimé n'est a priori pas réutilisée pour un nouvel enregistrement afin d'éviter une confusion

*IL PEUT DONC Y AVOIR DES VALEURS ABSENTES POUR UNE CLEF SI NUMÉRO,
EXEMPLE : 1 3 4 5 6 ... (2 SUPPRIMÉ)*

¹ La condition suivant le mot-clef `where` peut en fait être plus développée (cf. page 42) ; par ailleurs, si la commande est réduite à la forme `delete from table`, la table entière est alors effacée, vidée.

RECHERCHE DE DONNÉES AVEC PHPMYADMIN

RECHERCHE ASSISTÉE D'ENREGISTREMENTS

● RECHERCHE DANS UNE TABLE AVEC PHPMYADMIN

▪ EXEMPLE N° 1 : TROUVER TOUS LES LÉGUMES

Choisir la table puis utiliser l'onglet RECHERCHER

Indication de la question par comparaison(s) de colonne(s)

Colonne	Type	Interclassement	Opérateur	Valeur
numero	int(10)		=	
nom	varchar(40)	utf8_general_ci	LIKE	
type	enum('LEGUME', 'FRUIT')	utf8_general_ci	=	LEGUME
prix	decimal(7,2)		=	

+ Options

Exécuter

ici, les enregistrements avec le champ `type` valant `LEGUME`
et enfin valider avec le bouton EXÉCUTER



	numero	nom	type	prix
<input type="checkbox"/>	1	CAROTTE	LEGUME	1.00
<input type="checkbox"/>	2	BROCOLIS	LEGUME	2.00
<input type="checkbox"/>	3	PANAIS	LEGUME	3.00
<input type="checkbox"/>	5	CHOU DE CHINE	LEGUME	5.00

Le résultat est
présenté sous
la forme d'une
table

▪ EXEMPLE N° 2 : TROUVER TOUS LES NOMS DES FRUITS

Indication des colonnes à récupérer via clic sur OPTIONS

Options

Choisir les colonnes à afficher

numero
nom
type
prix

DISTINCT

type enum('LEGUME', 'FRUIT') = LEGUME

FRUIT



	nom
<input type="checkbox"/>	KIWI
<input type="checkbox"/>	POMME

UNE RECHERCHE DANS LA BASE RENVOIE UNE TABLE DE DONNÉES

RECHERCHE DE DONNÉES AVEC SQL : SELECT

RECHERCHE D'ENREGISTREMENTS DANS UNE TABLE DE LA BASE

▪ COMMANDE D'INTERROGATION « SELECT »

Syntaxes de la requête (versions simplifiées ¹)

```
select * from table
```

tous les enregistrements avec toutes les colonnes

```
select liste de colonnes from table
```

tous les enregistrements avec les seules colonnes ² indiquées dans la liste

```
select * from table
```

where *condition*

certains enregistrements avec toutes les colonnes

```
select liste de colonnes from table
```

where *condition*

certains enregistrements avec les seules colonnes indiquées dans la liste

▪ RESTRICTION DANS UNE INTERROGATION

Interrogation limitée aux enregistrements vérifiant une condition

Cas simple de condition : comparaison entre colonne et valeur

Syntaxe (version simplifiée) : where *colonne comparaison valeur*

▪ EXEMPLES

```
select * from produit
```

where type = 'LEGUME'

tous les légumes, avec toutes les colonnes de la table

```
select nom from produit
```

where type = 'FRUIT'

tous les noms de fruits

```
select nom, prix from produit
```

where prix >= 3

le nom et le prix des produits de prix au-moins égal à 3 €

```
select nom, telephone from personne
```

where code_postal = '75013'

nom et téléphone des personnes avec le code postal 75013

```
select ref_produit, quantite from vente
```

where date >= '2005-09-01'

référence et quantité des produits vendus à partir de septembre 2005

¹ D'autres formes plus complexes de la commande existent ; les principales seront présentées ci-après. Pour plus d'informations, voir l'aide-mémoire minimal du langage SQL de MySQL disponible sur le site de l'auteur.

² On peut aussi indiquer à la place d'une colonne, une expression correspondant à un calcul (voir page 44) ou une agrégation (voir page 47).

INTERROGATION EN SQL : RESTRICTIONS

POSSIBILITÉS DE SÉLECTION

▪ CONDITION DANS UNE RESTRICTION (where)

- une comparaison entre expressions ¹ avec les opérateurs
= > >= < <= <> (différence ²)
incluant la possibilité de comparer des dates
- la correspondance avec un texte générique ³ : `like 'motif'`
ou tout ce qui ne correspond pas au motif : `not like 'motif'`
où *motif* peut contenir « % » (aucun ou toute suite de caractères)
ou « _ » (un caractère quelconque)
- une combinaison logique :
`expression and expression` (« et », conjonction)
`expression or expression` (« ou », disjonction)
- la négation ⁴ : `not expression`
- l'appartenance ou non à un intervalle de valeurs (bornes comprises) :
`expression between valeurmin and valeurmax`
`expression not between valeurmin and valeurmax`
- la présence, ou l'absence, parmi une liste de valeurs :
`expression in (valeur1, valeur2, ...)`
`expression not in (valeur1, valeur2, ...)`
- existence ou non de la valeur :
`expression is null`
`expression is not null`

▪ EXEMPLES

```
select nom, prix from produit
where prix between 2 and 4.99
le nom et le prix des produits dont le prix est compris entre 2 € et 4,99 €
```

```
select nom, prix from produit
where type = 'LEGUME'
and (nom like '%CHOU%' or nom = 'BROCOLIS')
le nom et le prix des légumes qui sont un chou ou le brocolis
```

```
select nom from produit
where type = 'FRUIT' and nom not in ('POIRE', 'POMME')
le nom des fruits autres qu'une poire ou une pomme
```

¹ Les expressions sont présentées ci-après à la page 42

² MySQL permet aussi d'utiliser la notation spécifique « != » à la place de « <> »

³ La comparaison tient compte ou non de la casse des lettres (minuscule ou majuscule) ou des signes diacritiques (accent, cédille etc.) selon l'interclassement.

⁴ MySQL permet aussi d'utiliser la notation spécifique « ! » à la place du mot-clef « not ».

INTERROGATION EN SQL : EXPRESSIONS

PRINCIPALES EXPRESSIONS UTILISABLES DANS UNE CONDITION

▪ EXPRESSIONS

- le nom d'une colonne de la table
- une valeur (nombre, texte ou date)
- un calcul arithmétique entre deux expressions : + * - /
avec le parenthésage possible : ()
- la date du jour ou l'heure courante : `current_date` `current_time`
- une fonction de manipulation de texte :
 - `character_length(expression)` (nombre de caractères)
 - `lower(expression)` (passage en minuscules)
 - `upper(expression)` (passage en majuscules)
 - `trim(expression)` (élimination des espaces au début et à la fin)
 - `substring(expression from début)` (extrait à partir du caractère au rang *début*¹)
 - `substring(expression from début for taille)` (*taille* caractères à partir de *début*)

▪ EXEMPLES

```
select current_date  
la date courante
```

```
select numero, taille_initiale * prix from lot  
le numéro et la valeur en euros des lots selon le prix acheté au fournisseur
```

```
select numero, taille_reste * 100 / taille_initiale from lot  
where taille_reste > 0  
le numéro et la taille relative (en %) restante des lots non épuisés
```

```
select nom, telephone from personne  
where character_length(telephone) <> 14  
les nom et n° de téléphone des personnes dont ce n° n'a pas 14 caractères
```

```
select code_postal from personne  
where upper(ville) = 'PARIS'  
les codes postaux des personnes résidant à Paris, noté avec ou sans majuscules
```

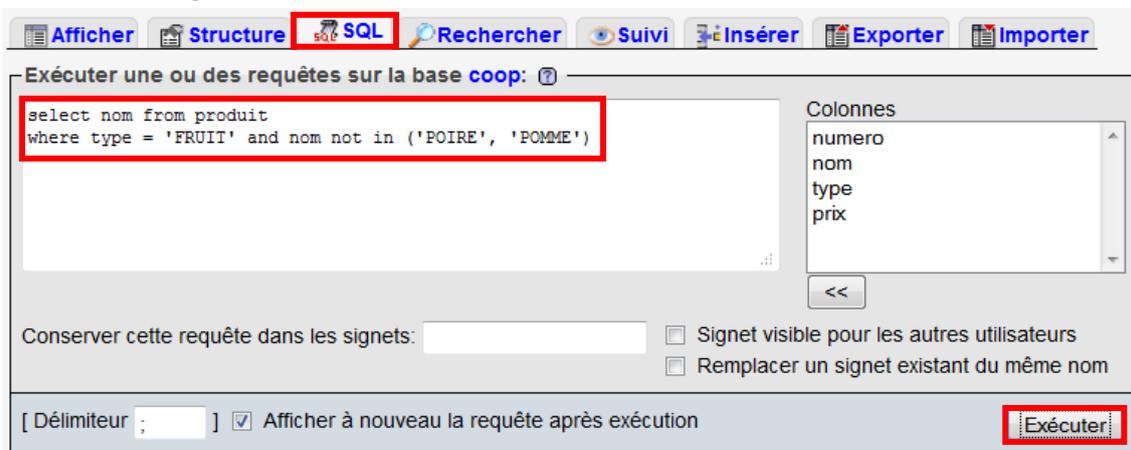
```
select upper(adress), code_postal, upper(ville) from personne  
where substring(code_postal, 1, 2) = '78'  
le nom avec l'adresse complète en majuscule des personnes des Yvelines (78)
```

¹ Le rang d'un caractère est compté à partir de 1.

INTERROGATION EN SQL AVEC PHPMYADMIN

EXEMPLE DE REQUÊTE DANS LA BASE coop

Utiliser l'onglet SQL

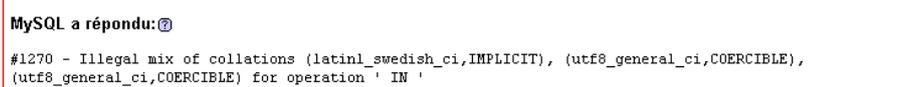


Taper la requête ¹ dans la zone puis cliquer le bouton EXÉCUTER



affichage de la requête et de la table en résultat

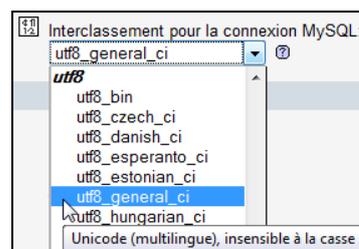
En cas d'anomalie « Illegal mix of collations »



1) Revenir à la page d'accueil de phpMyAdmin



2) A droite, régler la zone INTERCLASSEMENT POUR LA CONNEXION MYSQL à la valeur UTF8_GENERAL_CI



¹ Il n'y a pas à saisir les accents graves (« ` ») autour de noms de table ou de champ comme par exemple : `produit` ; cette notation est spécifique à MySQL et sert notamment à délimiter un nom composé de plusieurs mots (fortement déconseillé).

INTERROGATION EN SQL : DOUBLON, TRI, LIMITATION

COMPOSANTS COMPLÉMENTAIRES DE LA REQUÊTE « SELECT »

▪ VALEURS DISTINCTES

Élimination des doublons dans les résultats

`distinct` (mot à placer avant la liste des colonnes)

Exemple

```
select distinct prix from produit
tous les prix existants, mais sans répétitions
```

▪ TRI

Classement des résultats

`order by colonne` ou `order by colonne asc` (classement croissant)

`order by colonne desc` (classement décroissant)

`order by colonne1, colonne2, ...` (classements imbriqués)

(`order by` à placer vers la fin de la requête, après `where`)

Exemples

```
select nom, prix from produit
where type = 'LEGUME'
order by prix
le nom et le prix des légumes classés par prix croissant
select nom, prix from produit
order by type asc, prix desc
classement par type croissant, puis par prix décroissant
```

▪ LIMITATION

Limitation du nombre de résultats (spécifique à MySQL)

`limit taille` (seulement les *taille* premiers résultats)

`limit saut, taille` (*taille* résultats après la position *rang* ¹)

(`limit` à placer à la fin de la requête, après `order by` et `where`)

Exemples

```
select * from produit
limit 10
les 10 premiers produits
select nom, prix from produit
limit 10,20
le nom et le prix du 11ème au 30ème produits
```

¹ Attention, le premier résultat a pour rang zéro et non pas un !

INTERROGATION EN SQL : AGRÉGATION

CALCULS SUR TOUTES LES LIGNES OU PAR REGROUPEMENT

▪ FONCTIONS D'AGRÉGATIONS

Résultats correspondant à l'application d'une fonction de calcul sur les données renvoyées par l'interrogation

`count(*)` (comptage de tous les enregistrements)

`count(colonne)` (comptage des enregistrements où la colonne est définie)

`avg(colonne)` (moyenne, *average*)

`sum(colonne)` (somme)

`min(colonne)` ou `max(colonne)` (minimum ou maximum)

attention ! la parenthèse ouvrante doit être collée au nom de fonction ;
à placer seulement dans la liste après `select` mais pas après `where`

Élimination des doublons avant les calculs

ajout du mot `distinct` avant le nom de la colonne entre les parenthèses

Regroupement des résultats avant les calculs

`group by colonne1, colonne2, ...` (à placer juste après la restriction `where`)

Filtrage des résultats après les calculs

`having condition` (à placer juste après le regroupement `group by`)

Exemples

```
select count(*) from produit
  le nombre total de produits enregistrés
```

```
select count(distinct ville) from personne
  le nombre total de villes de résidence des personnes
```

```
select type, count(nom) from produit
group by type
  les types et le nombre total de produit (avec un nom défini) pour chaque type
```

```
select avg(prix) from produit where type = 'LEGUME'
  le prix moyen des légumes
```

```
select type, max(prix) from produit
group by type
  le prix maximal pour chaque type de produit
```

```
select type, min(prix) as min, max(prix) as max from produit
group by type
  minimum et maximum des prix pour chaque type de produit
```

```
select ref_produit, sum(quantite) as nb_total from vente
group by ref_produit
having nb_total > 100
  les références des produits vendus en quantité totale de plus de 100
```

INTERROGATION EN SQL : SURNOM, SOUS-RECHERCHE

COMPOSANTS COMPLÉMENTAIRES DE LA REQUÊTE « SELECT »

▪ DÉFINITION D'UN SURNOM DE COLONNE

Renommage d'une colonne ou nom affecté à un résultat correspondant à une fonction d'agrégation ou un calcul (dans la liste après `select`)
`as surnom` ou `surnom` (placé après nom de colonne, agrégation, calcul)
surnom réutilisable en filtrage (`having`) mais pas en restriction (`where`)

Exemples

```
select distinct type as genre from produit
```

liste des types de produit, dans une colonne appelé « genre »

```
select date, prix * quantite as total from vente order by date
```

liste des valeurs totales des ventes, dans une colonne appelé « total »

```
select type, count(nom) nb from produit group by type
```

les types et le nombre total de produit pour chaque type appelé « nb »

▪ UTILISATION D'UNE SOUS-RECHERCHE

une sous-recherche est une interrogation sans tri

une sous-recherche renvoyant une simple valeur peut figurer comme valeur dans une expression, notée entre parenthèses : (*interrogation*)

une sous-recherche renvoyant une table peut figurer comme table dans `from`, notée entre parenthèses suivi d'un surnom obligatoire : (*interrogation*) `as surnom` OU (*interrogation*) `surnom`

Exemples

```
select ref_produit, date, taille_initiale from lot
```

```
where taille_initiale =
```

```
(select max(taille_initiale) from lot)
```

lots dont la taille initiale est égale au maximum de tous les lots

```
select avg(total)
```

```
from
```

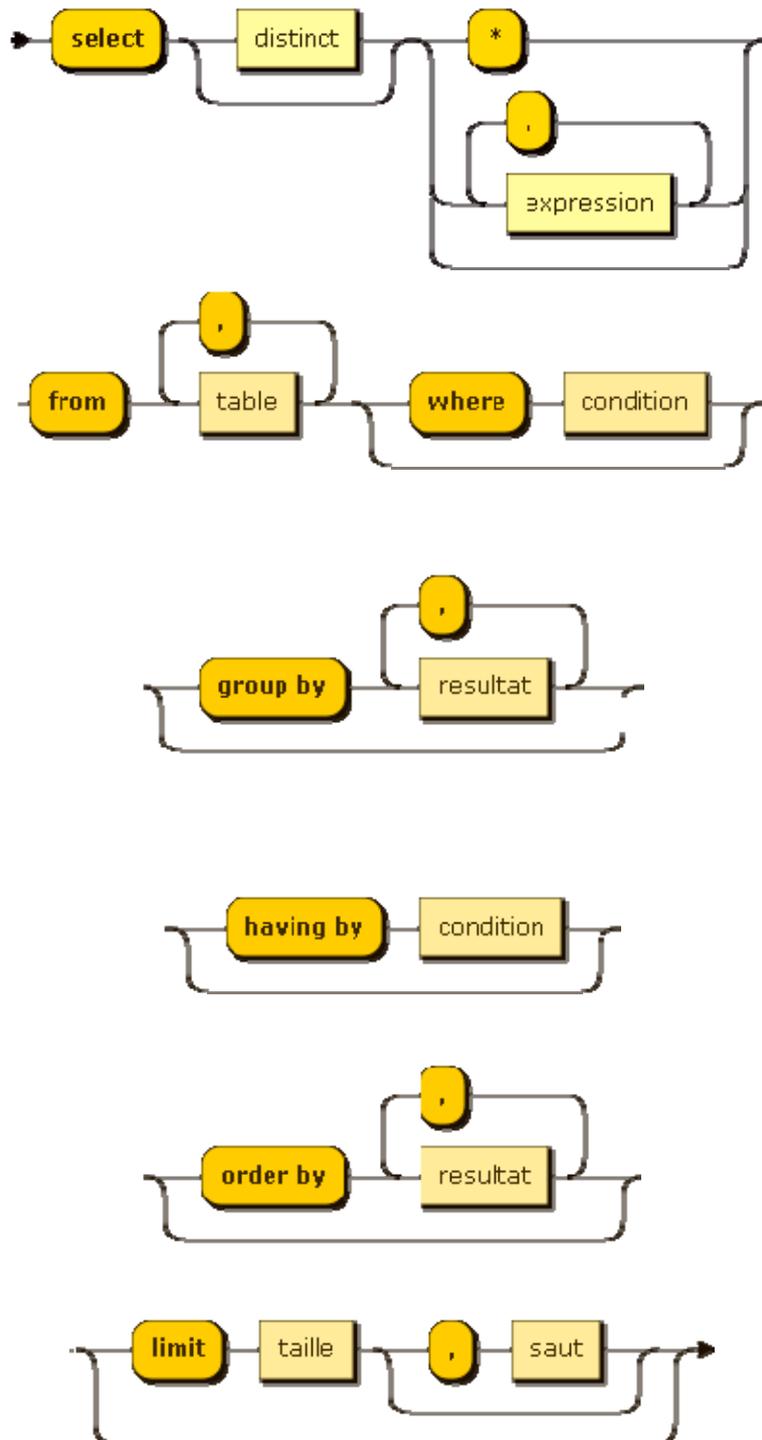
```
(select ref_produit, sum(quantite) as total
```

```
from vente group by ref_produit) as totaux_vente
```

moyenne des quantités totales de produits vendus

INTERROGATION EN SQL : FORME GÉNÉRALE

FORME GÉNÉRALE DE LA REQUÊTE « SELECT »



OPÉRATION DE JOINTURE

CAS D'INFORMATIONS RÉPARTIES DANS PLUSIEURS TABLES



EXEMPLE

Liste des numéros de lots avec date de vente et nom du vendeur

<i>lot</i>						
<u>numero</u>	ref_produit	ref_producteur	taille_initiale	taille_reste	prix	date
1	1	1	450	0	70	2005-12-29
2	3	2	100	0	300	2005-12-29
3	1	2	300	110	60	2005-12-30

+

<i>personne</i>					
<u>numero</u>	nom	adresse	code_postal	ville	telephone
1	DE LA RUE	9, rue Convention	93100	MONTREUIL	01.48.70.60.00
2	KING KING	1, place d'Italie	75013	PARIS	01.44.08.13.13
3	LILI	3, rue Résistance	95200	SARCELLES	01.34.38.20.00

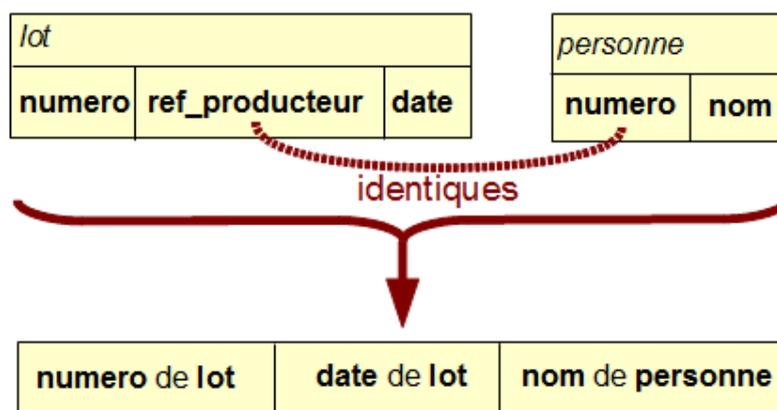


numero de lot	date de lot	nom de personne
1	2005-12-29	DE LA RUE
2	2005-12-29	KING KING
3	2005-12-30	KING KING

JOINTURE DE TABLE

ajouts de colonnes issues d'une autre table pour une même donnée

Jointure entre les tables « lot » et « personne »



LA JOINTURE PERMET D'ENRICHIR LES DONNÉES D'UNE TABLE

JOINTURE EN SQL

PREMIÈRE FORMULATION ¹

▪ SYNTAXE AVEC « JOIN »

Cas d'une jointure sur deux tables

```
select  liste de champs
  from  table1
  join  table2  spécification de jointure
```

Une spécification de jointure est

- soit dans le cas général : `on table1.colonne1 = table2.colonne2`

- soit en cas de champs de même nom : `using (colonne)`

c'est-à-dire équivalent à : `on table1.colonne = table2.colonne`

Nom de colonne préfixé par sa table afin d'éviter toute ambiguïté

(facultatif si aucune ambiguïté sur le nom de colonne) : `table.colonne`

Possibilité de rédéfinir le nom d'une table afin de simplifier l'écriture

`from table as nom` ou bien simplement `from table nom`

Cas général d'une jointure

```
select liste de champs
  from table1
  join table2 spécification2
  join table3 spécification3
  ...
```

▪ EXEMPLES

```
select lot.numero, lot.date, personne.nom
  from lot
  join personne on lot.ref_producteur = personne.numero
  jointure entre les tables lot et personne, sur le nom du producteur d'un lot
```

```
select lot.numero, lot.date,
  personne.nom as origine, produit.nom as nature
  from lot
  join personne on lot.ref_producteur = personne.numero
  join produit on lot.ref_produit = produit.numero
  jointure précédente complétée avec le nom du produit dans le lot
  et un nouvel intitulé pour les colonnes de nom du fournisseur et du produit
```

```
select l.numero, l.date, pe.nom as origine, pr.nom as nature
  from lot as l
  join personne as pe on l.ref_producteur = pe.numero
  join produit as pr on l.ref_produit = pr.numero
  jointure précédente avec abréviation des noms de table
```

¹ Cette formulation a été introduite en SQL dans la version de 1992 ; la présentation est ici simplifiée car elle permet en fait d'exprimer plusieurs formes de jointure.

JOINTURE EN SQL (SUITE)

ANCIENNE FORMULATION ¹

▪ SYNTAXE AVEC UNE CONDITION DE JOINTURE

Cas d'une jointure sur deux tables

```
select liste de colonnes
  from table1, table2
 where table1.colonne1 = table2.colonne2
```

Cas général d'une jointure

```
select liste de colonnes
  from table1, table2, ...
 where table1.colonne1 = table2.colonne2 and ...
```

Attention ! en cas d'indication d'un couple de tables sans restriction à la suite, on obtient le rajout à chaque ligne de la première table, de chaque ligne de la seconde table (« produit cartésien »)

```
select liste de colonnes from table1, table2
```

▪ EXEMPLES

```
select lot.numero, lot.date, personne.nom
  from lot, personne
 where lot.producteur = personne.numero
jointure entre les tables lot et personne, sur le nom du producteur d'un lot
```

```
select lot.numero, date, nom
  from lot, personne
 where producteur = personne.numero
écriture simplifiée de la jointure précédente
```

```
select lot.numero, lot.date,
  personne.nom as fournisseur, produit.nom as nature
  from lot, personne, produit
 where lot.producteur = personne.numero
 and lot.produit = produit.numero
jointure précédente complétée avec le nom du produit dans le lot
 et un nouvel intitulé pour les champs de nom du fournisseur et du produit
```

```
select l.numero, l.date,
  pe.nom as fournisseur, pr.nom as nature
  from lot as l, personne as pe, produit as pr
 where l.producteur = pe.numero and l.produit = pr.numero
jointure précédente avec abréviation des noms de table
```

¹ Cette formulation est la première apparue historiquement.

EXERCICES DE REQUÊTES SUR LA BASE « COOP »

DANS LE CADRE DU SYSTÈME D'INFORMATION DE LA COOPÉRATIVE :

1. PEUT-ON SUPPRIMER TOUS LES CHOUX EN UNE SEULE COMMANDE ?
2. QUELLE EST LA REQUÊTE POUR MULTIPLIER PAR 2 LE PRIX DES FRUITS ?
3. COMMENT OBTENIR LA LISTE TRIÉE DES NOMS, ÉCRITS EN MINUSCULE, POUR LES FRUITS DONT LE PRIX EST COMPRIS ENTRE 2 ET 3 EUROS ?
4. CALCULER LE NOMBRE ET LE POIDS TOTAL DES LIVRAISONS POUR CHAQUE PRODUIT
5. COMMENT OBTENIR LE NOM DES ACHETEURS DE CHOU DE CHINE ?
6. PEUT-ON CONNAÎTRE LES ACHETEURS DES LÉGUMES FOURNIS PAR KING KING ?

EXPORTATION DE LA BASE

MÉCANISME D'EXTRACTION DU CONTENU DE LA BASE (PHPMYADMIN)

● PRINCIPE

Création d'un fichier conservant une image de la base utile pour effectuer une sauvegarde complète de la base ou afin de réutiliser des données d'une ou plusieurs tables

● PROCÉDURE DE SAUVEGARDE COMPLÈTE

a) Sélectionner la base de données puis l'onglet EXPORTER

Server: localhost Base de données: coop

Structure SQL Rechercher Requête Exporter Importer Opérations Privilèges Supprimer

Schéma et données de la base

Exporter

Tout sélectionner / Tout désélectionner

lot
personne
produit
vente

CodeGen
 CSV
 CSV pour MS Excel
 Microsoft Excel 2000
 Microsoft Word 2000
 LaTeX
 Tableur "Open Document"
 Texte "Open Document"
 PDF
 SQL
 Texte Texy!
 XML
 YAML

Options

Commentaires mis en en-tête (l'n sépare les lignes)

Commentaires
 Utiliser le mode transactionnel
 Désactiver la vérification des clés étrangères
Mode de compatibilité SQL: NONE

Structure

Ajouter DROP TABLE / VIEW / PROCEDURE / FUNCTION / EVENT
 Ajouter IF NOT EXISTS
 Inclure la valeur courante de l'AUTO_INCREMENT
 Protéger les noms des tables et des champs par des ""
 Ajouter CREATE PROCEDURE / FUNCTION / EVENT

Inclure sous forme de commentaires

Dates de création/modification/vérification

Données

Insertions complètes
 Insertions étendues
Taille maximum de la requête générée: 50000
 Insertions avec délais (DELAYED)
 Ignorer les erreurs de doublons (INSERT IGNORE)
 Utiliser l'hexadécimal pour les BLOB
Type d'exportation: INSERT

Transmettre

Modèle de nom de fichier: DB_ (se souvenir du modèle)

Compression: aucune "zipé" "gzippé" "bzippé"

Exécuter

b) Vérifier les réglages automatiquement activés :

sélection de toutes les tables de la base,

EXPORTER à SQL, COMMENTAIRES, STRUCTURE, DONNÉES

c) Cocher TRANSMETTRE et choisir un mode de COMPRESSION :

AUCUNE si base petite, sinon "ZIPPÉ" en cas de gros volume

d) Cliquer sur le bouton EXÉCUTER et enregistrer le fichier produit ¹,

identifié avec le nom de la base et le suffixe SQL ; exemple : coop.sql

¹ Le fichier créé est au format de texte brut ; il contient toutes les commandes nécessaires à la reconstitution des tables de la base, exprimées dans le langage SQL.

IMPORTATION D'UNE BASE

MÉCANISME DE RÉCUPÉRATION D'UNE BASE SAUVEGARDÉE (PHPMYADMIN)

● PRINCIPE

Recréation d'une base précédemment sauvegardée à partir d'un fichier conservant une image de la base ou afin d'ajouter des données à une ou plusieurs tables

● PROCÉDURE DE REPRISE D'UNE BASE COMPLÈTE

a) Créer une base de données (vide)

b) Cliquer sur l'onglet IMPORTER

Serveur: localhost > Base de données: coop_double

Structure SQL Rechercher Requête Exporter Importer Opérations Privilèges Supprimer

Fichier à importer

Emplacement du fichier texte: easyphpcoop.sql (Taille maximum: 20 480 Kio)

Jeu de caractères du fichier: utf8

Ces modes de compression seront détectés automatiquement : aucune, gzip, bzip2, zip

Importation partielle

Permettre l'interruption de l'importation si la limite de temps est sur le point d'être atteinte. Ceci pourrait aider à importer des fichiers volumineux, au détriment du respect des transactions.

Nombre d'enregistrements (requêtes) à ignorer à partir du début: 0

Format du fichier d'importation

SQL

Options

Mode de compatibilité SQL: NONE

c) A EMPLACEMENT DU FICHIER TEXTE, indiquer le fichier contenant la sauvegarde de la base de données à l'aide du bouton PARCOURIR

d) Lancer l'importation via le bouton Exécuter

phpMyAdmin

Serveur: localhost > Base de données: coop_double

Structure SQL Rechercher Requête Exporter Importer Opérations Privilèges Supprimer

✓ L'importation s'est terminée avec succès, 15 requêtes exécutées.

```
-- phpMyAdmin SQL Dump
-- version 3.1.1
-- http://www.phpmyadmin.net
--
-- Serveur: localhost
-- Généré le : Dim 02 Janvier 2011 à 21:11
-- Version du serveur: 5.1.30
-- Version de PHP: 5.2.8
```

Base de données: _double

coop_double

- lot
- personne
- produit
- vente

En fin d'opération, affichage d'un compte-rendu synthétique et indication des tables importées

INDEX

- , 55
- !, 53
- !=, 53
- %, 53
- (), 55
- *, 55
- /, 55
- _, 36, 39, 53
- ` , 36, 57
- +, 55
- <, 53
- <=, 53
- <>, 53
- =, 53
- >, 53
- >=, 53
- 1 à plusieurs, 17
- 127.0.0.1, 33
- A_I, 37
- absence, 53
- accent grave, 36, 57
- agrégation, 60
- analyse des besoins, 27
- and, 53
- Apache, 34
- as, 62, 65
- asc, 59
- association, 8, 16, 19, 27
- attribut, 27
- attributs, 8
- auto_increment, 37, 39
- auto-increment, 47
- avg, 60
- base de donnée, 13
- base de données, 4
- between, 53
- cardinalité, 15, 17, 19, 27
- casse, 53
- chaîne de caractères, 41
- champ, 12
- character_length, 55
- classement, 59
- clef, 12, 24, 49
- clef étrangère, 19, 24, 44
- clef primaire, 19, 24
- comparaison, 51, 53
- comptage, 60
- concepteur, 46
- condition, 53
- contrainte, 27
- contrainte de clef, 24
- contrainte de domaine, 24
- contrainte d'intégrité référentielle, 24
- correspondance, 53
- count, 60
- create database, 36
- create table, 41
- current_date, 55
- current_time, 55
- date, 39, 41
- dec, 39
- decimal, 39
- default, 40
- défaut, 37
- définition d'un champ, 39
- delete, 49
- dépendance fonctionnelle, 23
- desc, 59
- différence, 53
- distinct, 59, 60
- domaine, 12, 24
- double precision, 39
- doublon, 59
- EasyPHP, 33, 34
- élimination des espaces, 55
- enregistrement, 12
- entier, 41
- entité, 8, 10, 27
- enum, 39
- espace, 41
- et, 53
- existence, 53
- exportation, 69
- expression, 55
- extraction de texte, 55
- filtrage, 60
- float, 39
- format, 8
- group by, 60
- having, 60
- identificateur, 10, 19, 27
- Illegal mix of collations, 57
- importation, 71
- in, 53
- incohérence, 24
- incrémentation automatique, 47
- index, 12, 44
- InnoDB, 43
- insert, 47
- insertion, 47
- int, 39
- integer, 39
- intégrité référentielle, 27, 43, 44
- interclassement, 53
- intervalle, 53
- is not null, 53
- is null, 53
- join, 65
- jointure, 64, 65, 67
- lettres accentuées, 36

like, 53
limit, 59
limitation, 59
liste, 41, 53
localhost, 33
lower, 55
majuscule, 55
max, 60
min, 60
minuscule, 55
mise à jour, 48
modèle conceptuel, 28
modèle entité-relation, 9
modèle relationnel, 13, 28
modélisation conceptuelle, 8
mot de passe, 35
Moteur de stockage, 43
moyenne, 60
MyISAM, 43
MySQL, 34
négation, 53
non existence, 53
not, 53
not in, 53
not like, 53
not null, 39
null, 39, 47
numeric, 39
n-uplet, 12
on, 65
or, 53
order by, 59
ou, 53
phpMyAdmin, 35
plusieurs à plusieurs, 15
préfixe, 65
présence, 53
primary key, 40
privilege, 35
produit cartésien, 67
recherche, 50
redondance, 23, 27
réel, 41
références, 46
regroupement, 60
relation, 12
requête, 31
restriction, 51
retour à la ligne, 41
root, 35
sauvegarde, 69
schéma, 46
select, 51, 63
sélection, 53
serveur, 4
SGBD, 31
show fields, 41
signes diacritiques, 53
smallint, 39
somme, 60
sous-recherche, 62
SQL, 31, 70
substring, 55
sum, 60
suppression, 49
surnom, 62, 65
système d'information, 3
système informatique, 4
table, 12, 16, 19
taille, 55
texte, 41
trait de souligné, 36
tri, 59
trim, 55
tuple, 12
unsigned, 39
update, 48
upper, 55
using, 65
valeur atomique, 8
valeur obligatoire, 27
valeurs distinctes, 59
varchar, 39
virgule, 41

SOMMAIRE

SYSTÈME D'INFORMATION	2
Introduction au système d'information	3
Système d'information : vue informatique	4
Système d'information : réalisation.....	5
Système d'information : vue pratique	6
PRINCIPES DES BASES DE DONNÉES	7
Modélisation des données.....	8
Donnée du produit.....	9
Donnée de la personne	10
Entités et base de données	11
Tables produit et personne.....	12
Donnée de la vente	13
Association vente et base de données.....	14
Donnée du lot	15
Lot dans la base de données	16
Association et base de données	17
Modélisation de la coopérative.....	18
Exercices d'extension du modèle	19
Redondance des données	20
Incohérence des données	21
Exercice sur la cohérence	22
Modélisation conceptuelle : récapitulatif.....	23
Passage à la base de données	24
Exercices sur la modélisation.....	25
UTILISATION DES BASES DE DONNÉES.....	26
Introduction.....	27
Cas de EasyPHP sous Windows.....	28
Réglages de EasyPHP.....	29
Outil phpMyAdmin	30
Création d'une base	31
Création d'une table avec phpMyAdmin.....	32
Création d'une table en SQL	33
Création d'une table en SQL (suite)	34
Intégrité référentielle en MySQL.....	35
Intégrité référentielle en MySQL (suite).....	36
Intégrité référentielle en MySQL (fin).....	37
Ajout de données	38
Mise à jour de données	39
Suppression de données.....	40
Recherche de données avec phpMyAdmin.....	41
Recherche de données avec SQL : select	42
Interrogation en SQL : restrictions.....	43
Interrogation en SQL : expressions	44
Interrogation en SQL avec phpMyAdmin.....	45
Interrogation en SQL : doublon, tri, limitation	46
Interrogation en SQL : agrégation	47
Interrogation en SQL : surnom, sous-recherche	48
Interrogation en SQL : forme générale	49
Opération de jointure.....	50
Jointure en SQL	51
Jointure en SQL (suite)	52
Exercices de requêtes sur la base « coop »	53
Exportation de la base.....	54
Importation d'une base	55
INDEX.....	56