

# Qu'est-ce qu'une base de données ?

# Sommaire

Introduction.....	3
<b>I. Un ensemble d'informations structurées.....</b>	<b>4</b>
A. Une base de données de plusieurs tables.....	5
B. Des tables de jointure pour les relations .....	7
C. Exemple d'optimisation du contenu... ..	8
D. ... et de la structure.....	9
E. Initiation au Modèle Conceptuel de Données .....	10

Crédits des illustrations :  
© Fotolia, Studio graphique.

## Les repères de lecture



Retour au chapitre



Définition



Objectif(s)



Espace Élèves



Vidéo /  
Audio



Point important /  
À retenir



Remarque(s)



Pour aller  
plus loin



Normes et lois



Quiz

```

connection($Host, $User, $Password);
mysql_connect($Host, $User, $Password);
Selection Db
mysql_select_db($dbname);

$sql = "select * from visiteurs where ((Salon = $id)";
$req = mysql_query($sql) or die('Erreur SQL !<br>'. $sql. '<br>');

header("Content-Type: application/txt-tab-delimited-table");
header("Content-disposition: attachment; filename=$id.txt");

if (mysql_num_rows($req) != 0) {
    // titre des colonnes
    $fields = mysql_num_fields($req);
    $i = 0;
    while ($i < $fields) {
        echo mysql_field_name($req, $i).";";
        $i++;
    }
    echo "\n";

    // données de la table
    while ($arrSelect = mysql_fetch_array($req, MYSQL_ASSOC)) {
        foreach ($arrSelect as $selem) {
            echo "$selem;";
        }
        echo "\n";
    }
    // Fermeture Connection MySql
    mysql_close();
}

```

## Introduction

Cette première partie vise à expliquer avec un peu de théorie ce que sont les bases de données et comment nous pouvons nous en servir.

Une base de données est finalement un simple outil logique de rangement (ou stockage) de l'information qui permet de créer des relations entre ces informations. Nous allons étudier les principes d'utilisation puis verrons comment optimiser le stockage des informations en fonction de leur nature et enfin comment élaborer un schéma nous permettant de comprendre une base de données. Nous n'effectuons donc pas de développement dans cette partie, mais manipulons uniquement des tableaux et des informations.

## I. Un ensemble d'informations structurées

Une base de données est un **ensemble d'informations structurées et organisées en tableaux**. Ces tableaux sont appelés **des tables**. Les informations peuvent être de diverses natures, en fonction de nos besoins, et sont rangées **par colonnes et par lignes**.

Les colonnes sont aussi appelées des champs ou des attributs et les lignes sont aussi nommées des **tuples** ou des enregistrements.

La colonne indique le type d'information, par exemple des couleurs ou prénoms et les lignes contiennent l'information : les couleurs (bleu, rouge, vert, etc.) ou les prénoms (Paul, Virginie, etc.). L'intérêt est que nous pouvons ainsi associer entre elles ces différentes informations.

Prenons par exemple la table suivante dans laquelle nous avons enregistré deux prénoms et leurs identifiants respectifs.

Tableau n°1

id	prenom
1	Paul
2	Virginie

Dans cet exemple qui contient donc deux lignes, les noms sont donc associés aux identifiants. Le prénom Paul correspond à l'identifiant 1 et le prénom Virginie correspond à l'identifiant 2. Inversement, l'identifiant 1 correspond au prénom Paul et l'identifiant 2 correspond au prénom Virginie. Nous avons donc associé ces identifiants et ces prénoms de manière précise.

La table contient ici deux champs mais peut naturellement en contenir beaucoup plus ce qui est très pratique si nous voulons enrichir notre contenu. Nous pouvons avoir besoin par exemple d'ajouter des informations supplémentaires comme les adresses email respectives de Paul et Virginie.

Tableau n°2

id	prenom	email
1	Paul	paul@email.com
2	Virginie	virginie@email.com

La table contient donc désormais les trois champs suivants :

- id;
- prenom;
- email.

Nous remarquons que les lignes nous permettent de trouver une adresse email grâce à l'identifiant. L'email [paul@email.com](mailto:paul@email.com) est, en effet, associé à l'identifiant 1 car ces deux informations sont enregistrées sur la même ligne.

Continuons à enrichir le contenu de la table et enregistrons par exemple le nom, l'adresse, le sexe et la couleur préférée de Paul et Virginie. Profitons-en pour rajouter deux lignes dont les identifiants respectifs seront donc 3 et 4.

Tableau n°3

id	prenom	email	nom	adresse	couleur	sexe
1	Paul	paul@email.tld	Henri	2, allée du roman 75018 Paris	bleu	Homme
2	Virginie	virginie@email.tld	Bernardin	3 rue de paradis 75010 Paris	rouge	Femme
3	Marguerite	marguerite@email.tld	Delatour	1 boulevard Saint-Martin 69005 Lyon	bleu	Femme
4	Étienne	etienne@email.tld	Delatour	1 boulevard Saint-Martin 69005 Lyon	vert	Homme

La table est désormais bien remplie : elle contient 7 champs et 4 enregistrements.

## A. Une base de données de plusieurs tables

Nous pouvons ajouter un très grand nombre de colonnes (ou champs) dans la table et cela nous permet donc de stocker un très grand nombre d'informations pour chaque ligne.

En revanche, et nous pouvons le constater ici, plus il y a d'informations, moins elles sont regroupées de manière logique et cohérente. Il n'y a, en effet, pas de rapport direct entre une adresse et une couleur.

Par ailleurs, nous pouvons nous rendre compte que des informations sont répétées dans cette table. Les identifiants 3 et 4 ont la même adresse, les identifiants 1 et 3 ont la même couleur préférée, les adresses contiennent des villes semblables, etc.

Il est donc pertinent de regrouper entre elles les informations de même nature. Pour ce faire, nous allons retirer des informations de cette table afin de les répartir dans deux nouvelles tables.

Au vu des types d'informations contenues, nous allons créer deux nouvelles tables « adresse » et « couleur » puis modifier la table initiale afin d'en retirer les champs adresses et couleurs tout en préservant la relation entre les personnes, leur adresse et leur couleur préférée.

Tableau n°4 Nouvelle table adresses

id	adresse
1	2 allée du roman 75018 Paris
2	3 rue de paradis 75010 Paris
3	1 boulevard Saint-Martin 69005 Lyon

Tableau n°5 Nouvelle table couleurs

id	couleur
1	bleu
2	rouge
3	vert

Tableau n°6 Table personnes modifiée.

id	prenom	nom	email	couleur_id	adresse_id
1	Paul	Henri	paul@email.tld	1	1
2	Virginie	Bernardin	virginie@email.tld	2	2
3	Marguerite	Delatour	marguerite@email.tld	1	3
4	Étienne	Delatour	etienne@email.tld	3	3

Nous avons remplacé les champs couleur et adresse par couleur\_id et adresse\_id qui vont contenir les identifiants respectifs des adresses et des couleurs de la table initiale.

Ainsi, nous possédons trois tables que nous avons nommées logiquement en fonction du type d'informations contenues, ces informations étant initialement contenues dans la première table de démonstration.

1. La table contenant les adresses a été nommée : adresses ;
2. La table contenant les couleurs a été nommée : couleurs ;
3. La table contenant les personnes a été nommée : personnes.

Nous nous rappelons que la table initiale permettait d'associer les personnes à leur adresse et couleur préférée respective, ce qui est toujours le cas dans notre nouvelle organisation.

Vérifions que Paul est bien associé à la couleur bleue.

L'information « Paul » est contenue dans la table personnes et a comme identifiant id : 1

L'information « bleu » est contenue dans la table couleurs et a comme identifiant id : 1

Ainsi, nous savons que l'identifiant de la personne id 1 (Paul) est relié à l'identifiant de la couleur couleur\_id 1 (bleu), conclusion : Paul est toujours associé à la couleur bleue qui est désormais enregistrée dans une autre table.

En ajoutant champs *couleur\_id* et *adresse\_id* dans la table « personnes », nous gardons ainsi les associations entre les identifiants des personnes et les identifiants des couleurs et des adresses.

Si nous voulons savoir quelle est la couleur préférée d'Étienne qui a l'identifiant 4, il suffit de regarder la ligne « 4 » et de constater que la couleur possède l'identifiant couleur\_id 3, ce qui, si on se reporte à la table couleurs, correspond à la valeur « vert ».

Ainsi, nous avons gardé les associations initiales tout en allégeant le contenu des tables.

Ce qui n'est pas du tout négligeable si on considère que notre table « personnes » peut contenir par exemple plusieurs dizaines de milliers d'enregistrements.

## B. Des tables de jointure pour les relations

Nous pourrions appréhender la question de la relation entre les informations de manière différente en créant une autre table dont le rôle spécifique serait d'enregistrer les associations entre les personnes et les couleurs.

Cette nouvelle table pourrait s'appeler *personne\_couleur* et contiendrait uniquement les associations des identifiants comme suit :

Table personnes\_couleurs

Tableau n°7

personne_id	couleur_id
1	1
2	2
3	1
4	3

Ce type de table est appelée une table de **jointure**, car elle contient uniquement la relation entre des informations. Dans le cas présent, elle relie l'identifiant d'une personne à l'identifiant d'une couleur.

Selon le même principe, nous pouvons également créer une table de jointure pour l'association des personnes et des adresses.

Dans ce cas, nous pourrions retirer les champs *couleur\_id* et *adresse\_id* de la table personnes puisque nous venons de voir que les associations sont bien présentes dans la table de jointure.

La table personnes serait donc plus simple :

Tableau n°8

id	prenom	nom	email
1	Paul	Henri	paul@email.tld
2	Virginie	Bernardin	virginie@email.tld
3	Marguerite	Delatour	marguerite@email.tld
4	Etienne	Delatour	etienne@email.tld

Comme nous le voyons, l'architecture de la base de données peut être établie de plusieurs manières différentes. **C'est un travail important que de bien réfléchir à la manière dont nous souhaitons que les informations soient organisées en préservant de manière pérenne leurs relations.** Les tables de jointure permettent les relations et simplifient les tables ; en revanche, leur mise en oeuvre lors des développements est plus complexe et nécessite une pratique PHP avancée.

Dans la suite de ce cours, nous préférons donc utiliser les tables contenant les champs avec les identifiants des informations enregistrés dans les autres tables.

## C. Exemple d'optimisation du contenu...

Si nous avons stocké toutes ces informations dans notre base de données, c'est que nous souhaitons pouvoir les utiliser. Nous pouvons, par exemple, avoir besoin de la liste des personnes qui habitent la ville de Lyon. Nous effectuons donc une recherche en nous aidant de la relation entre la table « personnes » et la table « adresses », mais celle-ci possède les adresses complètes et à chaque fois que nous effectuerons une recherche, il nous faudra du temps supplémentaire pour extraire les noms des villes. Puisqu'il s'agira vraisemblablement d'un besoin fréquent, il est alors souhaitable d'optimiser la table « adresses » en ajoutant les champs qui nous permettront de structurer plus finement les contenus de la table.

Nous allons alors décomposer les adresses en voirie, code postal et ville et créer les champs correspondants à ces types de contenu.

Tableau n°9 Table adresses modifiée.

adresse_id	voirie	code_postal	ville
1	2 allée du roman	75018	Paris
2	3 rue de paradis	75010	Paris
3	1 boulevard Saint-Martin	69005	Lyon

Nous pouvons ainsi effectuer plus rapidement notre recherche grâce au « champ ville » qui contient uniquement les noms de villes où habitent les personnes enregistrées dans la base.



## D. ... et de la structure

Si nous regardons attentivement la table après modification, nous remarquons que les noms des villes sont répétés plusieurs fois. En application de ce que nous avons déjà vu à propos des relations entre les personnes et les couleurs, nous pouvons ici encore optimiser la base mais cette fois en intervenant sur sa structure en créant une table villes destinée uniquement à contenir les noms de villes et éviter les répétitions inutiles d'informations.

Comme vu précédemment, le résultat sera donc l'obtention de deux tables « adresses » et « villes », la table « adresses » contenant l'identifiant *ville\_id* afin de préserver l'association voirie/ville.

Tableau n°10 Table adresses

id	voirie	code_postal	ville_id
1	2 allée du roman	75018	1
2	3 rue de paradis	75010	1
3	1 boulevard Saint-Martin	69005	2

Tableau n°11 Table villes

id	ville
1	Paris
2	Lyon

À ce stade, notre base de données contient déjà un nombre de tables important et nous ressentons le besoin d'effectuer un premier schéma nous permettant de visualiser d'un coup d'oeil les relations entre ces tables. Les relations entre les tables valorisent finalement les informations qui y sont contenues et il est important d'en avoir une vue d'ensemble.

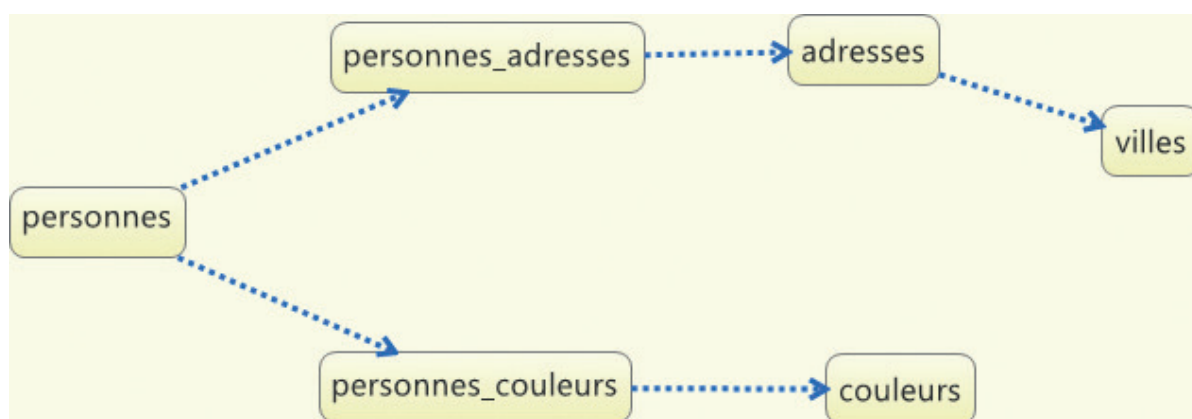


Fig. 1 Relation entre les bases de la table

Ce premier schéma permet d'appréhender la structure de la base et d'en constater les relations. Nous l'avons réalisé après avoir créé les tables mais nous aurions tout aussi bien pu le réaliser avant de créer nos tables et nous en servir comme un outil d'aide à la conception de la base de données que nous devons créer.

Un tel schéma pourrait également être communiqué à d'autres développeurs afin de réfléchir en équipe à la conception.

## E. Initiation au Modèle Conceptuel de Données

Le Modèle Conceptuel de Données (MCD) est un format de schéma qui détaille les bases de données. Il s'agit d'un **format standard** dans le monde du développement. Le MCD exprime la structure et les relations entre les tables de la base de données, nous parlons alors de modélisation. Il existe différents outils de modélisation (comme par exemple le MLD, modèle logique des données), mais le MCD a fait ses preuves comme outil et est encore très largement utilisé. Du reste, sa bonne compréhension permet d'aborder ensuite d'autres outils du même type.

Prenons un cas simple de base de données pour comprendre le MCD, la gestion d'une bibliothèque de prêt de livres, en nous limitant dans cet exemple aux informations basiques essentielles au fonctionnement de la bibliothèque.

Dans tout projet, avant de démarrer la réflexion sur la structure et le contenu de la base de données, il importe de bien comprendre de quelle manière nous allons l'utiliser par la suite. Nous déterminons donc les règles de gestion du projet, c'est-à-dire les différents mécanismes qui sont nécessaires au bon fonctionnement.

Dans le cas d'une bibliothèque, nous pouvons proposer les cinq règles de gestion basiques suivantes :

1. Les utilisateurs empruntent des livres.
2. Les utilisateurs sont enregistrés par leur prénom.
3. Les utilisateurs peuvent emprunter deux livres en même temps au maximum.
4. La bibliothèque référence ses ouvrages avec le titre du livre uniquement
5. Si l'utilisateur emprunte un livre, l'identifiant de l'utilisateur est ajouté dans la table livres. Donc si le livre est disponible, le champ utilisateur\_id est vide, sinon le livre est indisponible et ne peut pas être emprunté.

Ces règles de gestion étant énoncées, nous pouvons maintenant créer les tables de la base de données.

### 1. Élaboration des tables

#### a. Table utilisateurs

La table utilisateurs est structurée par le champ id et le champ prenom.



Ce cas est ici volontairement rendu très simple. Il conviendrait pour un tel projet de préciser les durées de prêt, d'ajouter des informations sur les utilisateurs et les livres, d'ajouter le nombre d'exemplaires, les auteurs etc.

Tableau n°12

id	prenom
1	Paul
2	Virginie
3	Marguerite

### b. Table livres

La table livres est structurée par le champ id, le champ titre, et le champ utilisateur\_id.

Tableau n°13

id	titre	utilisateur_id
1	PHP et MySQL	
2	Apprentissage du PHP	
3	Etude du PHP	

En appliquant strictement les règles de gestion utiles au fonctionnement de la bibliothèque, nous obtenons donc une base de donnée contenant deux tables reliées entre elles par l'identifiant utilisateur\_id.

Passons maintenant à la traduction de la base en schéma avec le Modèle Conceptuel de Données.

## 2. Élaboration du Modèle Conceptuel de Données

Le schéma ci-dessous est le MCD de la base que nous venons de concevoir.

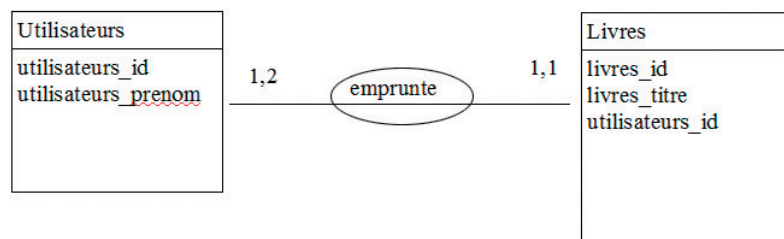


Fig. 2

Le MCD permet instantanément de visualiser les tables qui sont représentées par les deux blocs à gauche et à droite. Ces blocs sont intitulés avec le nom des tables et contiennent en-dessous la liste des champs. La simple lecture de ce schéma permet donc de savoir que la base de données contient les tables utilisateurs et livres et quels sont leurs champs respectifs.

Le MCD permet également de transcrire schématiquement les règles de gestion. Il montre ici que chaque utilisateur peut emprunter des livres en traçant un trait entre les deux tables et en écrivant la mention « emprunte » qui est entourée au milieu.

Cette mention qui précise le type d'action se traduit précisément au moyen des chiffres 1,2 et 1,1 (qu'il faut lire 1-2 et 1-1) qui sont situés au-dessus du trait à côté des tables.

Nous pouvons ainsi lire que **1** utilisateur peut emprunté **2** livres au maximum (**1,2**) et que **1** livre peut être emprunté par **1** seul utilisateur (1,1).

Si le nombre de livres pouvant être empruntés était illimité, nous aurions 1, N à côté de la table Utilisateurs, N signifiant un nombre indéfini.

Comme nous le voyons le MCD est un outil de formalisation, qui modélise la base de données et reformule les règles de gestions en un schéma standard.

Dans le cadre d'un développement, la réalisation du MCD est une tâche pouvant être complexe et elle est affectée généralement au développeur spécialisé dans les bases de données. Il ne semble donc pas utile dans le cadre de ce cours de s'exercer plus avant sur le MCD mais il est cependant important, en tant que développeur web ayant à travailler avec les bases de données, d'en connaître le principe d'élaboration et de s'en servir lorsque nous avons à concevoir une base de données, si besoin en effectuant un travail complémentaire de veille spécifique sur le sujet.

L'adresse suivante permet d'approfondir utilement la théorie des bases de données et la modélisation : <http://cyril-gruau.developpez.com/merise/>