I. Généralités

1) SGBD

➢ Une structure de base de données est une manière d'organiser les données pour les traiter plus facilement.

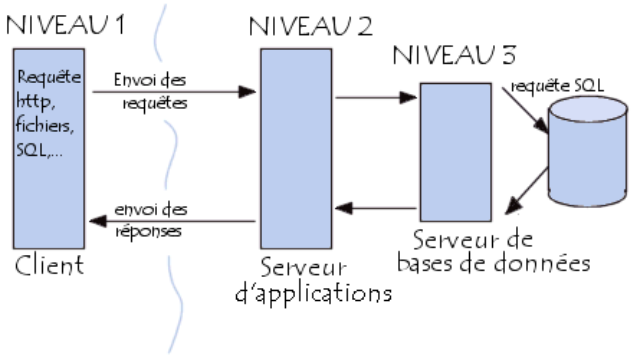
> Exemple 1 : Quand on veut ranger ses livres, on peut soit les mettre en vrac, soit les classer par nom d'auteur, soit par date de parution, etc.

➢ Un système de gestion de base de données ou SGBD (en anglais DBMS : database management system) « est un logiciel destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations » (source : Wikipédia).

Parmi les plus langages ou logiciels les plus répandus, on peut citer : MySQL (ou MariaDB), PostgreSQL, Libreoffice Base, DB Browser for SQLite pour les gratuits ; Oracle Database, IBM DB2, MS Access pour les payants.

2) Architecture trois-tiers

➢ L'architecture trois-tiers est un découpage en trois couches logicielles d'un SGBD.
Première couche : affichage et interaction avec l'utilisateur (exemple : distributeur de banque, page internet, application smartphone)
Deuxième couche : décrit le fonctionnement de l'entreprise, utilise les données pour les présenter de façon exploitable par l'utilisateur ; cette couche doit tenir compte des besoins de l'utilisateur et anticiper les demandes de celui-ci.
Troisième couche : accès aux données (qui peuvent être stockées de différentes manières, suivant les langages de programmation, à différents endroits, sur un réseau etc.)
Intérêt : le programmeur peut modifier une couche sans toucher aux autres.



Source : http ://www.commentcamarche.net/contents/221-reseaux-architecture-client-serveur-a-3-niveaux

3) Gestion des bases de données sur Internet

Les bases de données sont rarement manipulées à la main mais plutôt créées puis nourries en utilisant des logiciels. Sur internet, on utilise souvent :
➢ le langage PHP qui permet, en conjonction avec le programme MYSQL, de récupérer des données dans une base pour les présenter sur une page internet (c'est le cas pour de nombreux sites web) ou d'envoyer des données saisies par un internaute dans une base ;
➢ phpmyadmin, qui est un ensemble de pages PHP, permet de créer une base et de l'administrer assez simplement.

4) Bases et tables de données

a) Un exemple

Exemple 2 :
Une association culturelle chinoise propose à ses adhérents un système de prêt de livres pour l'année 2017–2018. Quand un élève emprunte un livre, son professeur complète un tableau (dans un tableur par exemple) qui commence ainsi (Nom_E et Prénom_E sont le nom et le prénom de l'emprunteur) :

N°	Titre	Auteur	Nom_E	Prénom_E	Tél.	Pris le	Rendu le	Activité	Prof
1	Le chinois en 2 h	Maître Chu	Hochon	Paul	0606060606	3–9	4–9	Chinois	Didier
2	Taiji Quan ou Qi Gong ?	Sue Pless	Térieur	Alain	0320032003	20–9	30–11	Taiji Quan	Michel
3	Les méridiens	F. Magellan	Nastic	Jim	1212121212	27–9		Qi Gong	Éric
4	Le plein de vide	B. Pascal	Néçante	Eva	0000000100	01–10		Méditation	Nadine
5	Le chinois pour les nuls	Maître Chu	Ochon	Paul	0706060606	5–10		Chinois	Didier
6	Tai Chi Chuan ou Qi Gong ?	Sue Pless	Térieur	Alex	0320032003	7–10	19–11	Taiji Quan	Luc

Cette représentation des données peut éventuellement convenir pour une association mais pas quand la masse de données est plus imposante (exemple : BDD topographique) pour plusieurs raisons :
– la saisie des données est laborieuse : il faut à chaque fois entrer le nom de l'emprunteur, celui de son professeur, son activité, etc.
– il peut y avoir des erreurs de saisie (exemple le nom et le numéro de tél. de M. Hochon, le titre du livre de Mme Pless, etc.) ; ce qui peut être très génant (par exemple pour le numéro de téléphone) ;
– les recherches d'informations (exemples : « qui a emprunté Taiji Quan ou Qi Qong ? » ou « combien de fois a-t-on emprunté les livres de Maître Chu ? ») sont laborieuses et les erreurs de saisie auront un impact sur les résultats de ces recherches ;
– toute modification doit être répercutée plusieurs fois, si M. Hochon change de numéro de téléphone, il faudra retrouver toutes les lignes où il apparaît (bien orthographié ou pas !) et changer à chaque fois son numéro…

L'idée est donc de faire en sorte que chaque information n'apparaisse qu'une seule fois en regroupant les données en sous-tableaux, ou tables, choisis avec pertinence.

Ici par exemple :

Activités	
id	Libellé
1	Taiji Quan
2	Qi Gong
3	Méditation
4	Cours de chinois

Professeurs		
id	Prénom	Id_Activité
1	Didier	4
2	Michel	1
3	Éric	2
4	Nadine	3
5	Luc	1

Adhérents				
id	Nom	Prénom	Tél	Id_Prof
1	Térieur	Alex	0320032003	5
2	Térieur	Alain	0320032003	2
3	Néçante	Eva	0000000100	4
4	Nastic	Jim	1212121212	3
5	Hochon	Paul	0606060606	1

Livres			
id	Titre	Auteur	Id_Activité
1	Le chinois pour les nuls	Maître Chu	4
2	Les méridiens	F. Magellan	2
3	Le plein de vide	B. Pascal	3
4	Taiji Quan ou Qi Gong ?	Sue Pless	1
5	Taiji Quan ou Qi Gong ?	Sue Pless	2
6	Le chinois en 2 h	Maître Chu	4

Emprunts				
Num	Id_Livre	Id_Adhérent	Pris_le	Rendu_le
1	6	5	3–9	4–9
2	4	2	20–9	30–11
3	2	4	27–9	
4	3	3	01–10	
5	1	5	5–10	
6	4	1	7–10	19–11

Par exemple, dans la table « Emprunts », l'emprunt n°5 concerne le livre d'identifiant 1 donc (table « Livres ») « Le chinois pour les nuls », emprunté par l'adhérent n°5 donc (table « Adhérents ») Paul Hochon.

On peut ensuite, en lisant la table « Adhérents », retrouver le numéro de téléphone de Paul Hochon ou l'identifiant de son professeur, qui est 1 donc (table « Professeurs ») c'est Didier : deux méthodes pour relancer M. Hochon qui tarde à rendre le livre emprunté…

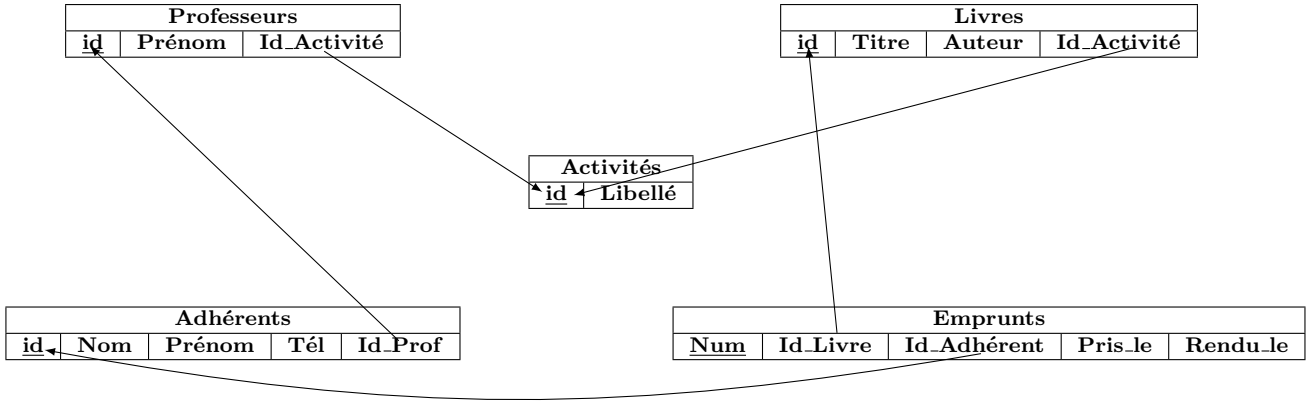
b) Vocabulaire

➢ chacun de ces tableaux est appelée table (ou relation) ;
➢ une table est organisée suivant des champs (ou attributs), par exemple, les champs de la table « Activités » sont « id » et « Libellé » ;
➢ les données correspond à un champ ont un certain type : par exemple, dans la table « Professeur », le champ « Prénom » est de type *texte*, tandis que le champ « id » est de type *nombre entier* ;
➢ chaque nouvelle ligne d'une table est une entrée (ou enregistrement ou occurrence) ;
➢ une base est un ensemble de tables : on pourra ici appeler la base « Association » par exemple ;
➢ clé primaire de recherche d'une table : un champ qui ne peut pas prendre deux fois la même valeur, souvent le champ « id » ; elle permet de faciliter les recherches dans la base ;
➢ clé étrangère d'intégrité : un champ d'une table qui fait référence à un champ d'une autre table, par exemple le champ « Id_Activité » de la table « Professeur », qui renvoie au champ « id » de la table « Activités ».

Remarque : avant de créer une base de donnée sur informatique, il est conseillé de prendre du papier et un crayon et d'écrire le schéma relationnel qui décrit les relations entre les différentes données (donc essentiellement les clés étrangères).

Exemple 2 :

Voici un schéma relationnel pour la base « association » :



En souligné, les clés primaires et les flèches signalent les liens entre les clés étrangères et les attributs qui y sont reliés.

Exemple 2 :

lisez cette documentation sur l'installation et l'utilisation du logiciel DB Browser for SQLite puis recréez la base de données de l'exemple 2 (vous pourrez appeler le fichier association.sqlite).

II. Requêtes SQL

La base de données étant constituée, nous allons pouvoir maintenant l'utiliser, c'est-à-dire l'interroger pour récupérer des informations. Cela passe par l'utilisation d'un langage, appelé SQL (Structured Langage Query). Par souci de clarté de lecture, nous noterons les mots-clés de ce langage en majuscule.

 Ouvrez dans DB Browser for SQLite votre base « association.sqlite » et testez les requêtes suivantes (les réponses sont ici données, vérifiez qu'il n'y a pas de problème).

1) Requête SELECT

a) Données non triées

La requête SELECT permet d'extraire des colonnes ou des lignes d'une table, suivant une certaine condition. La syntaxe la plus courante est : « SELECT *Champs* FROM *Table* WHERE *Conditions* »

Exemple 2 :
➢ La requête « SELECT Titre,Auteur FROM Livres » (sans condition) donnera comme résultat :

Titre	Auteur
Le chinois pour les nuls	Maître Chu
Les méridiens	F. Magellan
Le plein de vide	B. Pascal
Taiji Quan ou Qi Gong ?	Sue Pless
Taiji Quan ou Qi Gong ?	Sue Pless
Le chinois en 2 h	Maître Chu

➢ La requête « SELECT * FROM Professeurs » affichera la table « Professeurs » en entier.
➢ La requête « SELECT Prénom FROM Professeurs WHERE Id_Activité=1 » donnera comme résultat :

Prénom
Michel
Luc

➢ La requête « SELECT Titre FROM Livres WHERE Auteur=”Maître Chu” » (notez les apostrophes quand on fait une recherche sur un texte) donnera comme résultat :

Titre
Le chinois pour les nuls
Le chinois en 2 h

➢ On peut ne pas vouloir tous les résultats mais seulement quelques uns, pour cela on utilise la clause LIMIT ; ainsi la requête « SELECT Titre,Auteur FROM Livres LIMIT 3 » donnera comme résultat :

Titre	Auteur
Le chinois pour les nuls	Maître Chu
Les méridiens	F. Magellan
Le plein de vide	B. Pascal

(trois entrées).

Remarque : on peut préciser qu'on ne veut pas des premières lignes de la table avec la clause OFFSET, ainsi : « SELECT Titre,Auteur FROM Livres LIMIT 3 OFFSET 2 » donnera comme résultat :

Titre	Auteur
Le plein de vide	B. Pascal
Taiji Quan ou Qi Gong ?	Sue Pless
Taiji Quan ou Qi Gong ?	Sue Pless

(toujours trois entrées mais en oubliant les deux premières de la table).
➢ Certaines requêtes donnent des doublons (voir ci-dessus, le même titre apparaît deux fois) ; pour éviter cela, utiliser la clause DISTINCT ; « SELECT DISTINCT Auteur FROM Livres » donnera comme résultat :

Auteur
Maître Chu
F. Magellan
B. Pascal
Sue Pless

Nous verrons dans la suite comment écrire des requêtes plus complexes, du genre : lister les adhérents qui pratiquent le Taiji Quan et qui ont emprunté un livre entre le 1-9 et le 12-11.

b) Tri des données

La clause ORDER BY permet de trier les résultats en fonction d'un (ou de plusieurs) champ(s).

> Exemple 2 :
> ➢ La requête « SELECT Titre,Auteur FROM Livres ORDER BY Auteur » donnera :

Titre	Auteur
Le plein de vide	B. Pascal
Les méridiens	F. Magellan
Le chinois pour les nuls	Maître Chu
Le chinois en 2 h	Maître Chu
Taiji Quan ou Qi Gong ?	Sue Pless
Taiji Quan ou Qi Gong ?	Sue Pless

> ➢ La requête « SELECT Titre,Auteur FROM Livres ORDER BY Auteur DESC » fera un tri par ordre décroissant d'auteur.

c) Fonctions scalaires

On peut utiliser certaines fonctions pour transformer les résultats de requête. Par exemple :
➢ en utilisant UPPER, on convertit un texte en majuscule, comme dans la requête
« SELECT Titre,UPPER(Auteur) FROM Livres » où les noms des auteurs seront mis en majuscules ;
➢ en utilisant LENGTH, on récupère la longueur d'un texte, comme dans la requête
« SELECT Titre,LENGTH(Titre) FROM Livres » qui donnera comme résultat :

Titre	LENGTH(Titre)
Le plein de vide	16
Les méridiens	13
Le chinois pour les nuls	24
Le chinois en 2 h	17
Taiji Quan ou Qi Gong ?	23
Taiji Quan ou Qi Gong ?	23

d) Groupement et agrégation

Les fonctions suivantes permettent des calculs sur les données :
➢ COUNT(Champ) : donne le nombre d'entrées pour lesquelles la valeur de Champ n'est pas vide ; COUNT(DISTINCT Champ) fait la même chose en supprimant les doublons ;

> Exemple 2 : « SELECT COUNT(*) FROM Professeurs » donne le nombre d'entrées de la table Professeurs donc 5.
> « SELECT COUNT(DISTINCT Auteur) FROM Livres » donne le nombre d'auteurs distincts dans la table Livres donc 4.

➢ MAX(Champ) et MIN(Champ) donnent la plus grande et la plus petite valeur du champ,

> Exemple 2 : « SELECT MAX(Tél) FROM Adhérents » donne le plus grand numéro de téléphone (oui, c'est idiot) de la table Adhérents donc 1212121212.
> « SELECT MIN(Id_Activité) FROM Livres WHERE Auteur="Sue Pless" » (oui, c'est encore une requête inutile...) donne 1.

➢ SUM(Champ), AVG(Champ) et STD(Champ) : donnent la somme, la moyenne et l'écart type (« déviation standard ») des valeurs du champ.

> Exemple 2 :
>  Ajoutez un champ « Age » à la table « Adhérents ». Donnez un âge différent à chaque adhérent. Enfin, demandez l'âge moyen des adhérents : « SELECT AVG(Age) FROM Adhérents ».

Remarque : la fonction STD() existe en MySQL mais pas en SQLite. On peut utiliser par exemple : « SELECT SQRT((SUM(id*id) - SUM(id)*SUM(id)/COUNT(*))/COUNT(*)) from Professeurs » pour calculer l'écart type des identifiants de la table Professeurs.

➢ GROUP BY : cette clause permet de créer des regroupements de données.

Exemple 2 :
La requête « SELECT Auteur FROM Livres GROUP BY Id_Activité »
va créer, à partir des données, quatre groupes en fonction de Id_Activité mais une seule ligne est affichée pour chaque groupe !
On peut vérifier que les groupes existent bien ainsi :
« SELECT Auteur,Id_Activité,COUNT(*) FROM Livres GROUP BY Id_Activité »

➢ un regroupement ayant été fait avec GROUP BY, on peut effectuer un *post-traitement* avec la clause HAVING afin de ne retenir que les sous groupes qui vérifient une certaine condition.
La clause HAVING est analogue à WHERE sauf qu'elle s'applique à des groupes et pas à des lignes.

Exemple 2 :
La requête « SELECT Auteur FROM Livres GROUP BY Id_Activité HAVING COUNT(*) > 1 »
ne conserve que les groupes qui contiennent plus qu'un élément.

e) Jointure de plusieurs tables

Il est souvent nécessaire de faire des requêtes sur plusieurs tables en même temps.
Pour cela, on peut utiliser la syntaxe « JOIN ... ON ... » en reliant les différentes tables interrogées à l'aide des clés étrangères.
Pour accèder à un certain attribut d'une table, on utilise la notation *table.attribut*.
Par exemple, l'attribut « Nom » de la table « Adhérents » sera noté « Adhérents.Nom ».

Exemple 2 :
➢ La requête « SELECT Adhérents.Nom,Adhérents.Prénom, Professeurs.Prénom FROM Adhérents JOIN Professeurs ON Id_Prof =Professeurs.id » donnera comme résultat :

Nom	Prénom	Prénom
Térieur	Alex	Luc
Térieur	Alain	Michel
Néçante	Eva	Nadine
Nastic	Jim	Éric
Hochon	Paul	Didier

Remarques :
– le champ Id_Prof n'existe que dans la table Adhérents tandis qu'il y a un champ id dans différentes tables, pour préciser qu'on veut utiliser celui de la table Professeurs, on écrit donc Professeurs.id ;
– on peut aussi changer le nom des champs dans les tables (idA au lieu de id dans la table Activités) ;
– la requête « SELECT Adhérents.Nom,Adhérents.Prénom, Professeur.Prénom FROM Adhérents,Professeurs WHERE Id_Prof =Professeurs.id » donnerait le même résultat mais en prenant plus de temps ;
– on peut utiliser l'instruction AS qui permet de créer des alias, ce qui évite de taper plusieurs fois le nom complet d'une table ; par exemple : « SELECT a.Nom,a.Prénom, p.Prénom FROM Adhérents AS a JOIN Professeurs AS p ON Id_Prof =p.id ». À noter que certaines instructions, comme mysql_fetch_assoc() de PHP posent problème quand un même nom est utilisé plusieurs fois pour deux attributs (comme Prénom dans la requête précédente), rendant nécessaire l'utilisation des alias.

2) Les requêtes INSERT, DELETE, UPDATE, ALTER

Nous avons vu comment extraire des données d'une base, il faut savoir qu'il est évidemment possible d'en ajouter à la base avec la requête INSERT, d'en supprimer avec DELETE, d'en modifier avec UPDATE voire de modifier la structure d'une table avec ALTER (par exemple pour ajouter un champ à une table).
Ces requêtes ne sont pas au programme, mais j'ai inclus des liens en fin de ce cours si vous voulez aller plus loin.

III. Algèbre booléenne

➢ En logique classique (aristotélicienne), une proposition est soit vraie soit fausse.

1) Opérateur logique ET

➢ Soient A et B deux propositions. On appelle conjonction de A et de B la proposition A ET B qui est vraie quand A et B sont vraies en même temps.

> Exemple 3 :
> La proposition « Ceci est un cours de maths et d'informatique » est vraie si les deux propositions « Ceci est un cours de maths » et « Ceci est un cours d'informatique » sont toutes deux vraies.

➢ L'opérateur ET se note aussi & ou $\wedge$.

Remarque : on retrouve cette notion en probabilité lorsque qu'on écrit l'événement $A \cap B$, intersection des événements A et B.

➢ La table de vérité de l'opérateur ET est (si on note V pour « vrai » et F pour « faux ») :

A \ B	V	F
V	V	F
F	F	F

> Exemple 4 :
> La proposition A ET B : « Il pleut et j'ai pris mon parapluie » est fausse :
> – s'il ne pleut pas (A est faux) ;
> – s'il pleut mais que j'ai n'ai pas pris mon parapluie (A est vrai mais B est faux).

Remarque : si on note 1 pour « vrai » et 0 pour « faux » alors on peut remarquer que la table de vérité de l'opérateur ET revient à une multiplication : $1 \times 1 = 1$; $1 \times 0 = 0$, etc.

2) Opérateur logique OU

➢ Soient A et B deux propositions. On appelle disjonction de A et de B la proposition A OU B qui est vraie quand au moins une des propositions A et B est vraie.

> Exemple 5 :
> La proposition « Ceci est un cours de maths ou d'informatique » est vraie si au moins une des deux propositions « Ceci est un cours de maths » et « Ceci est un cours d'informatique » est vraie.

➢ L'opérateur OU se note aussi || ou $\vee$.

➢ On retrouve cette notion en probabilité lorsque qu'on écrit l'événement $A \cup B$, réunion des événements A et B.

➢ On parle ici d'un OU *inclusif* et pas d'un OU *exclusif* (dans la phrase « fromage ou dessert », il s'agit d'un OU exclusif : on ne peut pas choisir les deux).

➢ La table de vérité de l'opérateur OU est :

A \ B	V	F
V	V	V
F	V	F

Exemple 6 :
La proposition A OU B : « Il pleut ou j'ai pris mon parapluie » est fausse s'il ne pleut pas (A est faux) et si je n'ai pas pris mon parapluie (B est faux).

3) Propriétés algébriques

Considérons trois propositions A, B, C.

a) Commutativité

➢ ET est un opérateur commutatif :
La proposition A ET B est vraie si et seulement si la proposition B ET A est vraie.
On peut noter : (A ET B) $\iff$ (B ET A) (ou (A & B) $\iff$ (B & A)).

Exemple 7 : dire « J'ai bien mangé et il fait beau » revient à dire « Il fait beau et j'ai bien mangé ».

➢ OU est un opérateur commutatif :
(A OU B) $\iff$ (B OU A) (qu'on peut noter : (A $\vee$ B) $\iff$ (B $\vee$ A)).

b) Associativité

➢ ET et OU sont des opérateurs associatifs ; c'est-à-dire que l'on a :

$$(A \,\&\, B) \,\&\, C \iff A \,\&\, (B \,\&\, C)$$
$$(A \vee B) \vee C \iff A \vee (B \vee C)$$

c) Distributivité

➢ Distributivité de « ou » par rapport à « et » :

$$(P \vee (Q \,\&\, R)) \iff ((P \vee Q) \,\&\, (P \vee R))$$

➢ Distributivité de « et » par rapport à « ou » :

$$(P \,\&\, (Q \vee R)) \iff ((P \,\&\, Q) \vee (P \,\&\, R))$$

4) Négation

a) Définition

Soit A une proposition. La négation de A, notée NON A ou $\bar{A}$ est la proposition qui est VRAIE si et seulement si A est FAUX.

Exemple 8 :
La proposition contraire de A : « Il fait chaud » est $\bar{A}$: « Il ne fait pas chaud » (et pas « Il fait froid » !)

b) Négation d'une conjonction ou d'une disjonction

➢ La négation d'une conjonction est la disjonction des négations : $\boxed{\overline{A \,\&\, B} = \overline{A} \vee \overline{B}}$

Exemple 9 :
La négation de « Il pleut et j'ai pris mon parapluie » est « Il ne pleut pas ou je n'ai pas pris mon parapluie » (et pas « Il ne pleut pas et je n'ai pas pris mon parapluie » !).

➢ La négation d'une disjonction est la conjonction des négations : $\boxed{\overline{A \vee B} = \overline{A} \,\&\, \overline{B}}$

Exemple 10 :
La négation de « Les élèves pratiquant le foot ou le tennis auront cette année une médaille » est « Les élèves pratiquant le foot et ceux pratiquant le tennis n'auront pas de médaille cette année ».

5) Applications aux requêtes SQL

Exemple 2 :
➢ La requête « SELECT Prénom FROM Adhérents WHERE Nom=”Térieur” AND Id_Prof=5 » donnera comme résultat :

Prénom
Alex

➢ pour chercher les livres empruntés qui traitent de chinois ou qui ont été empruntés entre le 1er septembre et le 30 septembre, on peut utiliser la requête :
« SELECT Titre FROM Livres JOIN Emprunts ON Emprunts.Id_Livre=Livres.id
WHERE (Livres.Id_Activité=4) OR ((Emprunts.Pris_le>”1_9”) AND (Emprunts.Pris_le<”30_9”)) »
qui donnera :

Titre
Le chinois en 2 h
Taiji Quan ou Qi Gong ?
Les méridiens
Le chinois pour les nuls

IV. Quelques liens utiles

On pourra lire les documents suivants :
➢ Un bon diaporama de présentation.
➢ Les types de données.
➢ Les requêtes SELECT, page 1 et page 2.
➢ Insérer des données dans une base (INSERT).
➢ Supprimer ou modifier des données dans une base (DELETE et UPDATE).
➢ Modifier une base (ALTER).
➢ Bases de données et Python.
➢ memento Mysql.
➢ À propos de phpmyadmin, qui permet d’administrer une base de donnée de site internet : lien 1 et lien 2