

# Chapitre 2

## Bases de données : Pratique



Grâce aux bases théoriques acquises au chapitre précédent, vous voilà prêts à gérer vos premières bases de données.

Le SGBDR que nous utiliserons est **MySQL**, très connu, très utilisé et facile d'accès. Il est largement suffisant pour la majorité des projets que vous mènerez (gestion d'utilisateurs sur un site WEB, forum, données GPS...)

## 2.1 Installation de MySQL

MySQL s'utilise depuis une interface graphique ou directement en lignes de commandes depuis un terminal. Dans un soucis de simplicité, nous l'utiliserons dans un premier temps depuis l'interface graphique.

Cette interface graphique est accessible depuis un serveur WEB. Ainsi, en plus d'installer MySQL, il vous faudra installer un serveur WEB si vous souhaitez utiliser MySQL depuis chez vous.

### Windows / MAC

L'idéal est de télécharger un package complet comprenant :

- MySQL et phpMyAdmin (interface graphique)
- Apache (serveur WEB) et interpréteur PHP (langage WEB)

Exemples de programmes :

- WampServer (Windows) : <https://www.wampserver.com/en/>
- MAMP (Windows / MAC) : <https://www.mamp.info/en/downloads/>

### Linux

Pour Linux, l'installation se fait en 3 étapes :

- installation de MySQL
- installation de Apache (serveur WEB)
- installation de phpMyAdmin

On pourra se référer au lien suivant :

<https://itexpert.tips/fr/mysql-fr/installer-phpmyadmin-sur-ubuntu-linux/>

## 2.2 Interface graphique phpMyAdmin

### 2.2.1 Connexion à l'interface

Une fois le serveur Web en marche et *phpMyAdmin* installé, il suffit de se rendre à l'adresse :

`http://localhost/phpmyadmin`

`localhost` signifie que le serveur auquel on se connecte est le serveur local de la machine.

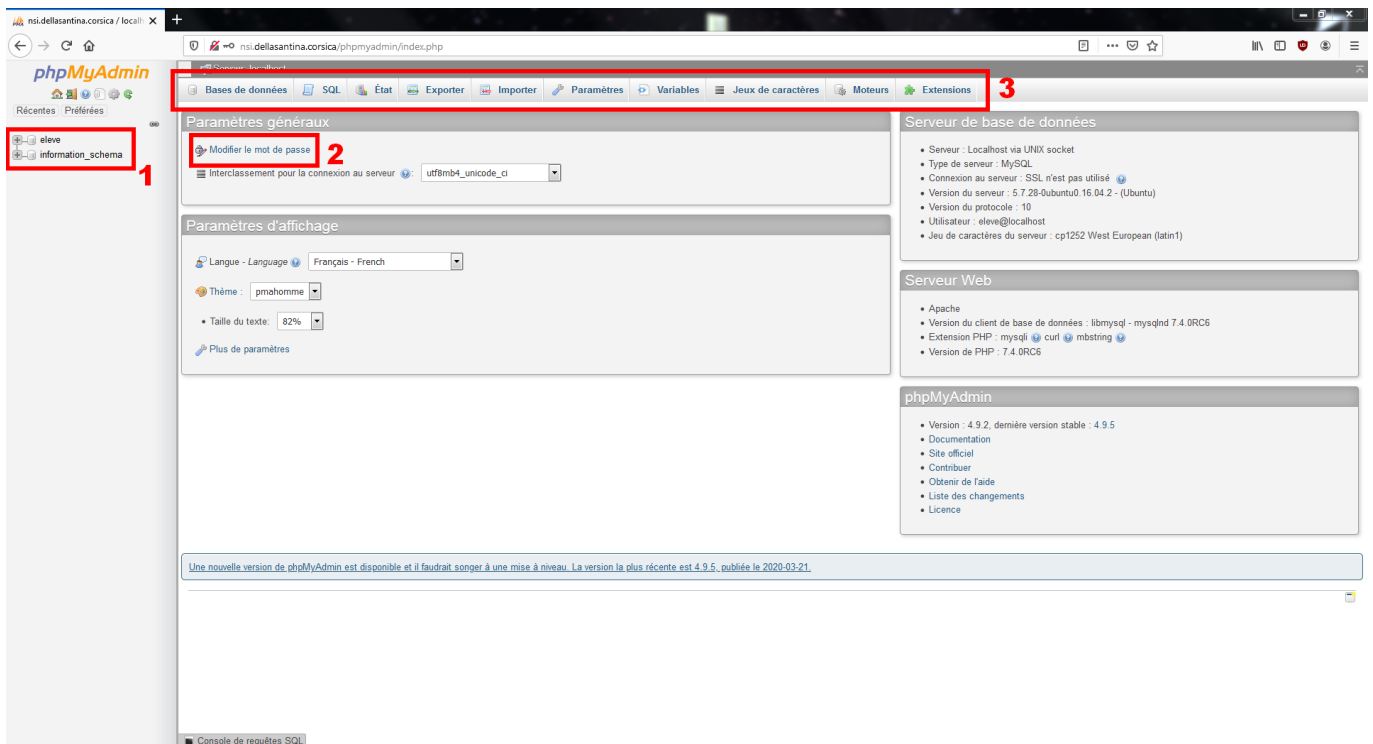
Il est tout à fait possible d'utiliser *phpMyAdmin* à distance, en renseignant l'adresse du serveur en lieu et place de `localhost`.

Par exemple, `http://192.168.1.37/phpmyadmin` ou encore `https://monsupersite.com/phpmyadmin`.

Dans le cadre de ce cours, vous utiliserez *MySQL* depuis le serveur du site WEB *python.dellasantina.corsica*, en suivant l'adresse :

`http://python.dellasantina.corsica/phpmyadmin`

Des codes d'accès vous sont gracieusement distribués par votre professeur.



Une fois connecté, vous vous retrouvez sur la page d'accueil de *phpMyAdmin*, depuis laquelle vous pouvez :

1. Parcourir les différentes bases de données qui vous sont accessibles (chaque élève dispose d'une base de données)
2. Modifier votre mot de passe
3. Parcourir les différentes possibilités de *phpMyAdmin* via le menu haut

### 2.2.2 Création de tables

Considérons les tables suivantes :

#### auteurs

<b>id</b>	UNSIGNED INT NOT NULL PRIMARY KEY
<b>nom</b>	VARCHAR(100) NOT NULL
<b>prenom</b>	VARCHAR(100) NOT NULL
<b>date_naissance</b>	DATE NOT NULL
<b>date_mort</b>	DATE

#### livres

<b>id</b>	UNSIGNED INT NOT NULL PRIMARY KEY
<b>id_auteur</b>	UNSIGNED INT NOT NULL
<b>titre</b>	VARCHAR(100) NOT NULL
<b>annee_parution</b>	SMALLINT NOT NULL

**Question 01** Quelles sont les clés primaires utilisées sur ces tables ?

**Question 02** Pourquoi la contrainte NOT NULL n'est pas utilisée pour le champ date\_mort de la table auteurs ?

Pour créer une table, il suffit de sélectionner une base de données. Une nouvelle page s'ouvre, il ne vous reste qu'à entrer le nom de la table et choisir le nombre de colonnes.

Nouvelle table

Nom:  Nombre de colonnes:

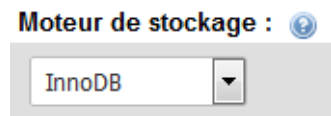
Il faut ensuite nommer un à un les champs utilisés, en précisant le type et les différentes contraintes :

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	Index	A_I
id	INT		Aucun(e)		UNSIGNED	<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
nom	VARCHAR	100	Aucun(e)			<input type="checkbox"/>	---	<input type="checkbox"/>
prenom	VARCHAR	100	Aucun(e)			<input type="checkbox"/>	---	<input type="checkbox"/>
date_naissance	DATE		Aucun(e)			<input type="checkbox"/>	---	<input type="checkbox"/>
date_mort	DATE		NULL			<input checked="" type="checkbox"/>	---	<input type="checkbox"/>

**Quelques précisions**

- Seules les colonnes jusqu'à *A\_I* nous intéressent
- On précise la clé primaire dans la colonne *Index* en sélectionnant PRIMARY
- *A\_I* signifie *Auto Increment* : on choisit cette option pour la clé primaire (et seulement pour elle), de sorte que MySQL incrémente automatiquement pour nous la valeur de *id* à chaque nouvel enregistrement
- Dans la table *auteurs*, la colonne *Taille/Valeurs* est seulement renseignée pour les VARCHAR
- Il n'y a pas de case NOT NULL, mais une case NULL à cocher (non cochée = NOT NULL)
- On peut préciser une valeur par défaut, comme pour la date de mort à NULL par défaut
- La colonne *Interclassement* est laissée vide : dans ce cas, le codage utilisé pour les caractères est identique à celui utilisé sur la table (*utf8\_general\_ci* dans ce cas)

On veille également que le **moteur de stockage** soit *InnoDB*, ce qui nous permettra de définir des clés étrangères.



Notre table est alors créée. On peut y accéder en la sélectionnant depuis la liste des tables, la parcourir ou afficher sa structure.



Notre table ne contient encore aucun enregistrement :

**id** **nom** **prenom** **date\_naissance** **date\_mort**

Mais sa structure est parfaitement définie :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	<b>id</b>	int(10)		UNSIGNED	Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	<b>nom</b>	varchar(100)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
3	<b>prenom</b>	varchar(100)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
4	<b>date_naissance</b>	date			Non	Aucun(e)			Modifier Supprimer Plus
5	<b>date_mort</b>	date			Oui	NULL			Modifier Supprimer Plus

**Question 03** Créer de même la table *livres*.

### 2.2.3 Création des clés étrangères

Avant d’insérer nos données, il est nécessaire de définir les relations entre nos tables, et donc les clés étrangères.

Pour désigner un champ comme étant une clé étrangère, il faut en premier lieu l’**indexer**.

Lorsqu’on crée un index sur un champ, MySQL enregistre et met à jour automatiquement une **liste ordonnée** des valeurs de ce champ. L’accès à ces données est alors beaucoup plus rapide.

☞ Une clé primaire est nécessairement indexée. C’est pourquoi on trouve l’attribut `PRIMARY` dans la colonne `Index` lors de la création de la table...

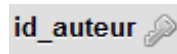
Pour définir un index, plusieurs choix s’offrent à nous. Une fois la structure de la table affichée, on peut :

1. Sélectionner `Index` dans la colonne `Actions` de la structure de la table
2. Cocher le champ à indexer et sélectionner `Index`
3. Créer un index depuis le menu dédié

The screenshot shows a database management tool interface. At the top, there are navigation tabs: Parcourir, Structure, SQL, Rechercher, Insérer, Exporter, Importer, Opérations, Suivi, Déclencheurs. Below these are buttons for 'Structure de table' and 'Vue relationnelle'. A table structure is displayed with columns: #, Nom, Type, Interclassement, Attributs, Null, Valeur par défaut, Commentaires, Extra, and Action. The table has four rows: 1. id (int(10), UNSIGNED, Non, Aucun(e), AUTO\_INCREMENT), 2. id\_auteur (int(10), UNSIGNED, Non, Aucun(e)), 3. titre (varchar(100), utf8\_general\_ci, Non, Aucun(e)), 4. annee\_parution (smallint(6), Non, Aucun(e)). The 'Action' column for each row contains 'Modifier', 'Supprimer', and 'Plus'. A red box highlights the 'Plus' button for the 'id\_auteur' row, with a red '1' next to it. Below the table, there are buttons for 'Tout cocher', 'Avec la sélection : Parcourir', 'Modifier', 'Supprimer', 'Primaire', 'Unique', 'Index', 'Texte entier', and 'Ajouter à'. The 'Index' button is highlighted with a red box and a red '2'. Below this is a section for adding columns to the index, with 'Ajouter', a dropdown showing '1', 'colonne(s)', 'après annee\_parution', and an 'Exécuter' button. Below that is a detailed view of the index being created, with columns: Action, Nom de l'index, Type, Unique, Compressé, Colonne, Cardinalité, Interclassement, Null, and Commentaire. The index is named 'PRIMARY', has type 'BTREE', is 'Unique' (Oui), 'Compressé' (Non), on the 'id' column, with a cardinality of '0' and an interclassement of 'A'. The 'Null' column is 'Non'. At the bottom, there is a button 'Créer un index sur' followed by a dropdown showing '1', 'colonnes', and an 'Exécuter' button, which is highlighted with a red box and a red '3'.

**Question 04** Créer un index simple (*Index*) sur le champ `id_auteur` .

Une petite clé apparaît pour signifier que l’indexation est effective.



L’index *Unique* permet d’indexer un champ en spécifiant l’unicité de chacune des valeurs pour ce champ.

**Question 05** Pourquoi est-ce une mauvaise idée de créer un index *Unique* sur ce champ ?

Une fois l’index créé, on se rend dans la **vue relationnelle** :



Contraintes de clé étrangère

Actions		Propriétés de la contrainte		Colonne	Contrainte de clé étrangère (INNODB)		
				Base de données	Table	Colonne	
Nom de la contrainte	ON DELETE	RESTRICT	ON UPDATE	id_auteur	eleve	auteurs	id
				+ Ajouter une colonne			

+ Ajouter une contrainte

**Question 06** Créer la contrainte de clé étrangère sur le champ `id_auteur` comme ci-dessus.

**Quelques précisions**

- Il n'est pas nécessaire de fournir un nom à la contrainte : MySQL le renseignera automatiquement
- `ON DELETE RESTRICT` signifie que lors de la suppression d'un utilisateur, la présence de son `id` dans la table *livres* créera une restriction : la suppression de l'utilisateur sera impossible
- `ON UPDATE CASCADE` signifie que lors de la mise à jour de l' `id` d'un auteur dans la table *auteurs* (ce qui n'arrive normalement pas...), alors tous les `id_auteur` correspondants dans la table *livres* seront mis à jour avec la nouvelle valeur
- `ON DELETE CASCADE` supprimerait le livre si son auteur est détruit : ce n'est pas forcément la meilleure idée!

La contrainte `ON DELETE SET NULL` permet de passer `id_auteur` à `NULL` si l'auteur est supprimé de la base de données.

**Question 07** Modifier la contrainte de clé étrangère avec la contrainte `ON DELETE SET NULL` . Expliquer le résultat.

**2.2.4 Insertion de données**

L'insertion des données se fait depuis l'onglet *Insérer*.



Depuis la table *auteurs*, on peut donc insérer un auteur :

Colonne	Type	Fonction	Null	Valeur
id	int(10) unsigned	<input type="text"/>		<input type="text"/>
nom	varchar(100)	<input type="text"/>		<input type="text"/>
prenom	varchar(100)	<input type="text"/>		<input type="text"/>
date_naissance	date	<input type="text"/>		<input type="text"/>
date_mort	date	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="text"/>

**Exécuter**

**Quelques précisions**

- Il n'est pas nécessaire de renseigner `id` : grâce à l'option *Auto Increment*, MySQL va déterminer lui même sa valeur
- Les champs de type `DATE` dispose d'une aide à la saisie (le petit calendrier)
- Par défaut, `date_mort` est égal à `NULL` : la case idoine est cochée par défaut

**Question 08** Remplir les tables *auteurs* et *livres* avec les données suivantes :

**auteurs**

id	nom	prenom	date_naissance	date_mort
1	Tolkien	John Ronald Reuel	1882-01-03	1973-09-02
2	Rowling	Joanne	1965-07-31	NULL
3	Lovecraft	Howard Philip	1890-08-20	1937-03-15
4	Martin	George Raymond Richard	1948-09-20	NULL

**livres**

id	id_auteur	titre	annee_parution
1	2	Harry Potter à l'école des sorciers	1998
2	2	Harry Potter et la Chambre des secrets	1999
3	2	Harry Potter et le Prisonnier d'Azkaban	1999
4	2	Harry Potter et la Coupe de Feu	2001
5	1	Le Seigneur des Anneaux	1954
6	3	L'Appel de Cthulhu	1926
7	1	Le Silmarillon	1977
8	4	Le Trône de fer	1996
9	1	Le Hobbit	1937

**Question 09** Que se passe-t-il si on ajoute un livre dont l' `id_auteur` est égal à 5 ?

**Question 10** Ajoutez un ou plusieurs auteurs ainsi que des livres associés.

## 2.2.5 Modification d'une table

Une idée surgit dans votre esprit : pourquoi ne pas ajouter les pays d'origine de chacun des auteurs ?

▷ ***Il fallait y penser avant !***

« C'est pas faux. »

En effet, il faut en général éviter de modifier la structure d'une table lorsque celle-ci contient déjà des données. Cela peut créer des erreurs inattendues.

**Exemple.** Un champ `email` de type `VARCHAR(100)` limite la taille maximale des adresses emails enregistrées à 100 caractères. « C'est beaucoup » se dit un programmeur naïf, qui s'empresse de changer le type pour un `VARCHAR(50)`. Malheureusement pour lui, une centaine d'adresses dépasse les 50 caractères. Ces adresses sont donc tronquées, et disparaissent définitivement dans le néant.

☞ *En réalité, MySQL va empêcher une telle situation d'arriver. C'est tout l'intérêt d'un SGBD.*

Nous allons modifier notre table `auteurs` en rajoutant un champ `pays` après le champ `prenom`. Pour faire cela, on se place dans la structure de la table et on choisit « Ajouter ... colonnes(s) après ... ».

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	int(10)		UNSIGNED	Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	nom	varchar(100)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
3	prenom	varchar(100)	utf8_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
4	date_naissance	date			Non	Aucun(e)			Modifier Supprimer Plus
5	date_mort	date			Oui	NULL			Modifier Supprimer Plus

Ajouter 1 colonne(s) après date\_mort Exécuter

**Question 11** Ajouter un champ `pays` après l'attribut `prenom` de la table `auteurs`.

On choisira une valeur par défaut égale à `NULL` : tous les enregistrements seront de valeur `NULL` pour le pays.

Vous réfléchissez un peu, et vous vous dites qu'il serait préférable de créer une table `liste_pays` contenant tous les pays du monde, définis par un identifiant unique. On pourrait alors créer une relation entre les tables `auteurs` et `pays`.

**Question 12** Modifier l'attribut `pays` en conséquence :

- Changer son nom pour `id_pays`
- Changer son type pour `UNSIGNED INT`

## 2.2.6 Import et export de données

### Import de données

En y réfléchissant bien (vous réfléchissez beaucoup), il est assez long de dresser une liste de tous les pays du monde. Il faudrait ajouter les pays un par un, et avant tout chercher une liste de tous ces pays...

Rassurez-vous, quelqu'un l'a déjà fait pour vous. Vous n'avez plus qu'à vous servir.

**Question 13** Rendez-vous sur la page <https://sql.sh/514-liste-pays-csv-xml> et téléchargez la liste des pays au format SQL (bas de la page).

SQL est un langage de requête structurée (*Structured Query Language*) servant à exploiter les bases de données relationnelles. Le fichier que vous venez de télécharger va en quelque sorte demander à MySQL de créer une table contenant la liste des pays inscrits à l'intérieur (n'hésitez pas à ouvrir le fichier avec un éditeur de texte).

Il ne reste qu'à l'importer depuis phpMyAdmin. Il vous suffit de vous placer dans votre base de données et de choisir l'option *Importer*.



**Question 14** Importer le fichier téléchargé dans votre base de données.

Si tout s'est bien passé, vous vous retrouvez désormais avec une nouvelle table nommée *pays* contenant 241 enregistrements.

	id	code	alpha2	alpha3	nom_en_gb	nom_fr_fr
<input type="checkbox"/> Éditer Copier Supprimer	1	4	AF	AFG	Afghanistan	Afghanistan
<input type="checkbox"/> Éditer Copier Supprimer	2	8	AL	ALB	Albania	Albanie
<input type="checkbox"/> Éditer Copier Supprimer	3	10	AQ	ATA	Antarctica	Antarctique
<input type="checkbox"/> Éditer Copier Supprimer	4	12	DZ	DZA	Algeria	Algérie
<input type="checkbox"/> Éditer Copier Supprimer	5	16	AS	ASM	American Samoa	Samoa Américaines
<input type="checkbox"/> Éditer Copier Supprimer	6	20	AD	AND	Andorra	Andorre
<input type="checkbox"/> Éditer Copier Supprimer	7	24	AO	AGO	Angola	Angola
<input type="checkbox"/> Éditer Copier Supprimer	8	28	AG	ATG	Antigua and Barbuda	Antigua-et-Barbuda

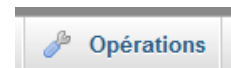
**Question 15** Une clé primaire est-elle définie sur cette table ? Si oui, quel est l'attribut correspondant ?

**Question 16** Des index sont-ils définis sur cette table ? Si oui, lesquels ?

**Question 17** Une valeur égale à *ABC* peut-elle être présente dans la colonne *alpha2* ?

Arrangeons un peu la table fraîchement importée.

**Question 18** Renommer la table en *liste\_pays* depuis l'onglet *Opérations*.



**Question 19** Supprimer les attributs `code`, `alpha3` et `nom_en_gb` de la table *liste\_pays*.

Il est temps de créer notre clé étrangère sur l'attribut `id_pays` de la table `auteurs` et de lier ainsi nos deux tables.

**Question 20** Créer un index simple sur l'attribut `id_pays` de la table `auteurs`.

**Question 21** Créer la contrainte de clé étrangère associée.

☞ Pour créer la contrainte, il est nécessaire de réaliser 2 modifications :

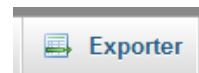
1. changer le moteur de stockage de la table `liste_pays` (Opérations  $\Rightarrow$  Moteur de stockage  $\Rightarrow$  InnoDB)
2. Modifier le type du champ `id_pays` de la table `auteurs` afin qu'il corresponde au type de `id` dans la table `liste_pays`

**Question 22** Modifier tous les enregistrements de la table `auteurs` afin de faire correspondre `id_pays` avec l' `id` de la table `liste_pays`. (On cherchera sur les internets les pays d'origine de chacun des auteurs)

### Export de données

Notre base de données est déjà bien remplie. Que diriez-vous de la sauvegarder ?

Pour faire cela, on va l'**exporter** depuis l'onglet du même nom.



**Question 23** Sélectionnez votre base de données et cliquez sur *Exporter*.

**Question 24** Laissez les options par défaut et cliquez sur *Exécuter* en bas de la page.

Enregistrez le fichier ainsi généré où bon vous semble.

**Modèles d'exportation :**

---

Nouveau modèle :

Modèles existants : Modèle : -- Sélectionner un modèle --

**Méthode d'exportation :**

Rapide, n'afficher qu'un minimum d'options

Personnalisée, afficher toutes les options possibles

**Format :**

SQL

☞ Par défaut, tout ce qui caractérise notre base est exporté : structure, données, clés étrangères... On peut décider des options d'export en choisissant une méthode d'exportation personnalisée.

☞ Par défaut, le format d'export est SQL. Il est néanmoins possible d'exporter des données en format CSV, ce qui permet de les importer depuis un tableur.

### 2.2.7 Suppression de données

Une fois votre base de données exportée, vos données sont sauvegardées dans un fichier SQL. On peut ainsi découvrir les opérations de suppression sans crainte.

**Question 25** Que se passe-t-il si on tente de supprimer des données dans la table *auteurs*? et dans la table *liste\_pays*?

**Question 26** Dans quelle table peut-on supprimer des enregistrements sans craindre d'erreur?

**Question 27** Supprimer tous les enregistrements de cette table :

- Sélectionner la table
- Cliquer sur *Opérations*
- Cliquer sur *Vider la table*

**Question 28** Supprimer de même tous les enregistrements de la table *auteurs*.

**Question 29** Supprimer la table *liste\_pays* :

- Sélectionner la table
- Cliquer sur *Opérations*
- Cliquer sur *Supprimer la table*

**Question 30** Supprimer de même les tables *auteurs* et *livres*.

☞ *Votre base de données est vide : passons à la suite!*

### 2.2.8 Sélection de données

Insérer et supprimer des données est intéressant, mais le principal intérêt des bases de données réside dans le **traitement** de informations qu'elles contiennent. Pour illustrer les possibilités de MySQL, nous allons utiliser des jeux de données plus conséquents.

**Question 31** Télécharger le fichier SQL suivant et importez le dans votre base de données :

<http://nsi.dellasantina.corsica/documents/BDD/NSI.sql>

**Question 32** Donner le schéma relationnel des 3 tables importées.

Vous savez déjà afficher les enregistrements d'une table depuis l'onglet *Parcourir*. Il est également possible d'effectuer une recherche dans une table avec l'onglet *Rechercher*.

**Question 33** Cliquez sur l'onglet *Rechercher* depuis la table *communes*.

Un formulaire de recherche apparaît, permettant la sélection et la projection des données (sur une ou plusieurs colonnes) :

Rechercher « par valeur » (joker : « % »)

Colonne	Type	Interclassement	Opérateur	Valeur
code_commune	int(3)		=	
code_departement	varchar(2)	utf8_general_ci	LIKE	
nom_commune	varchar(25)	utf8_general_ci	LIKE	
population	int(5)		=	
code_insee	varchar(5)	utf8_general_ci	LIKE	
latitude	decimal(12,10)		=	
longitude	decimal(12,11)		=	

Options

**Sélection**

Sélectionner les colonnes (au moins une) :

- code\_commune
- code\_departement
- nom\_commune
- population
- code\_insee
- latitude
- longitude

DISTINCT

Ou Critères de recherche (pour l'énoncé « where ») :

Nombre de lignes par page : 25

Ordre d'affichage :  Croissant  Décroissant

**Projection**

On peut par exemple chercher les communes de plus de 1000 habitants grâce à la ligne *population* :

population    int(5)    >=    1000

Le résultat apparaît rapidement :

✓ Affichage des lignes 0 - 24 (total de 58, traitement en 0.0014 seconde(s).)

```
SELECT * FROM 'communes' WHERE 'population' >= 1000
```

1 > >> |  Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette tab | Trier sur l'index: Aucun(e)

+ Options

	code_commune	code_departement	nom_commune	population	code_insee	latitude	longitude
<input type="checkbox"/> Éditer Copier Supprimer	1	2A	Afa	3080	2A001	41.9844089346	8.79828936731
<input type="checkbox"/> Éditer Copier Supprimer	4	2A	Ajaccio	69378	2A004	41.9347926638	8.70132275974
<input type="checkbox"/> Éditer Copier Supprimer	6	2A	Alata	3219	2A006	41.9735186682	8.73133206502
<input type="checkbox"/> Éditer Copier Supprimer	8	2A	Albitreccia	1649	2A008	41.8604746410	8.87771179182
<input type="checkbox"/> Éditer Copier Supprimer	9	2B	Aléria	2341	2B009	42.1059105058	9.49564551200
<input type="checkbox"/> Éditer Copier Supprimer	17	2A	Appietto	1734	2A017	42.0032023283	8.73355643393
<input type="checkbox"/> Éditer Copier Supprimer	32	2A	Bastelicaccia	3895	2A032	41.9352135675	8.83650751371

On voit notamment (ligne verte) qu'il y a 58 résultats (donc 58 communes de plus de 1000 habitants).

On peut aller plus loin :

- On peut omettre les champs `code_insee` , `latitude` et `longitude` en les dé-sélectionnant (onglet projection)
- Trier par population décroissante (onglet *Ordre d'affichage*)

	code_commune	code_departement	nom_commune	population
<input type="checkbox"/> Éditer Copier Supprimer	4	2A	Ajaccio	69378
<input type="checkbox"/> Éditer Copier Supprimer	33	2B	Bastia	44070
<input type="checkbox"/> Éditer Copier Supprimer	247	2A	Porto-Vecchio	11844
<input type="checkbox"/> Éditer Copier Supprimer	42	2B	Borgo	8381
<input type="checkbox"/> Éditer Copier Supprimer	37	2B	Biguglia	8026
<input type="checkbox"/> Éditer Copier Supprimer	96	2B	Corte	7499
<input type="checkbox"/> Éditer Copier Supprimer	120	2B	Furiani	5914

**Question 34** Grâce à l'outil de recherche, déterminer :

1. La commune la moins peuplée de Corse
2. Le nombre de communes de Haute-Corse
3. La dernière commune de Corse-du-Sud dans l'ordre alphabétique

**Recherche avec LIKE**

Le mot clé `LIKE` permet de rechercher une portion de texte ou de nombre. On l'utilise avec un *joker*, le caractère %, qui représente « n'importe quoi ».

Par exemple, la chaîne de caractères `A%` représente n'importe quelle chaîne de caractères composée d'un A suivi de... n'importe quoi. On peut ainsi sélectionner les communes de Corse commençant par A :



The screenshot shows a search interface with the following elements: a column name 'nom\_commune', a data type 'varchar(25)', an encoding 'utf8\_general\_ci', a search operator dropdown menu set to 'LIKE', and a search input field containing 'A%'.

☞ La recherche est **insensible à la casse**, donc `A%` et `a%` retournent les mêmes résultats

**Question 35** Grâce à l'outil de recherche, déterminer :

1. Une commune de Corse commençant par un D
2. Les communes de Corse terminant par *ccio*
3. Le nombre de communes de Corse dont le nom est composé (contient un -)

On se place maintenant dans la table *utilisateurs*.

**Question 36** Grâce à l'outil de recherche, déterminer :

1. Le nombre d'utilisateurs possédant une adresse mail sur le site WEB *supermail.fr*
2. L'âge du plus vieil utilisateur
3. Le nombre d'utilisateurs habitant à Ajaccio

**Jointures**

La dernière requête de la question précédente nécessite l'emploi d'une **jointure**, opération que nous avons découverte dans le chapitre précédent. Malheureusement, il n'est pas possible de faire de jointure avec l'onglet Recherche.

Grâce aux jointures, on pourrait notamment :

- Déterminer les utilisateurs d'une commune donnée
- Déterminer la commune où habite le plus grand nombre d'amateurs de pêche
- Déterminer les utilisateurs présents dans un rayon de 15km autour d'Ajaccio

Pour faire des jointures avec MySQL, il est nécessaire de manipuler le **langage SQL**.

Ça tombe bien, c'est justement le sujet du prochain chapitre !

**2.2.9 Bonus : le schéma relationnel de notre base**

**Question 37** Afficher depuis l'onglet *Concepteur* le schéma relationnel correspondant à notre base de données.