

TRAME

Objectifs :

Etre capable de comprendre comment fonctionne une BDD MariaDB.

Etre capable de configurer une BDD MariaDB.

Matériel à avoir :

Une machine Debian 10 sur laquelle vous avez préalablement installé LAMP, ainsi qu'une machine cliente Windows pour accéder à la base de données par l'interface de Adminer.

Consignes :

Dans un premier temps, vous allez rechercher des commandes MariaDB sur internet, pour connaître leur fonctionnement et leur syntaxe. Ensuite, dans un second document, d'exercice, en suivant les instructions, vous utiliserez ces commandes pour mettre en place une base de données simple afin de tester réellement les commandes et leurs syntaxes.

Table des matières

Se connecter à MariaDB	3
Gestion de la BDD en elle-même	3
Visualiser le nom des BDD existantes	3
Créer une base de données	3
Créer un utilisateur administrateur de cette base de données	4
Utiliser une base de données	4
Supprimer une BDD entière	4
Gestion de la structure des tables	4
Visualiser le nom des tables existantes dans la BDD	4
Visualiser la structure d'une table existante dans la BDD	5
Créer une table avec ses colonnes et leurs caractéristiques	5
Modifier une table (rajouter, supprimer et modifier une colonne)	6
Rajouter une colonne dans une table	6
Modifier une colonne dans une table	6
Supprimer une colonne dans une table	7

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Supprimer une table entière	8
Gestion des données contenues dans une table.....	9
Visualiser les enregistrements présents dans une table	9
Visualiser tous les enregistrements dans une table avec toutes les colonnes.....	9
Visualiser certains enregistrements dans une table avec toutes les colonnes	10
Visualiser tous les enregistrements dans une table avec des colonnes spécifiques	10
Insérer une ligne de données (un enregistrement) dans une table avec toutes les valeurs.....	11
Insérer une ligne de données (un enregistrement) dans une table en spécifiant uniquement les valeurs à remplir.....	12
Insérer plusieurs lignes dans une table en une seule commande.....	12
Modifier une donnée spécifique dans une ligne spécifique	13
Supprimer une donnée spécifique dans une ligne spécifique	13
Supprimer une ligne d'enregistrement entière	14
Éléments complémentaires.....	15
Les différentes structures conditionnelles possibles.....	15
Listes des types de données que vous pouvez utiliser sur les colonnes	15
Listes des contraintes/options que vous pouvez utiliser sur les colonnes	16
Clé primaire	18
Clé étrangère	20
Mise en place d'une clé étrangère à la création de la table	20
Mise en place de la clé étrangère avec modification d'une table existante.....	22
Table d'association.....	25
Insertion des enregistrements avec les clés étrangères.....	29

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Travail à faire :

Chercher les commandes MariaDB et leurs syntaxes pour :

Se connecter à MariaDB

En tant que super admin : `sudo mariadb`

En tant qu'utilisateur avec droits sur une BDD : `mariadb -u nomuser -p`

Gestion de la BDD en elle-même

Visualiser le nom des BDD existantes

Contexte : Vous voulez visualiser le nom de toutes les BDD existantes auxquelles vous avez accès avec votre utilisateur connecté.

`SHOW DATABASES ;`

```
MariaDB [exemple_bdd]> SHOW DATABASES;
+-----+
| Database |
+-----+
| exemple_bdd |
| information_schema |
+-----+
2 rows in set (0.000 sec)
```

Créer une base de données

`CREATE DATABASE nomBDD ;`

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Créer un utilisateur administrateur de cette base de données

```
GRANT ALL ON nomBDD.* TO 'nomuser'@'localhost' IDENTIFIED BY
'azerty' WITH GRANT OPTION;

FLUSH PRIVILEGES ;
```

Utiliser une base de données

```
USE nomBDD;
```

Supprimer une BDD entière

Contexte : Vous avez créé une BDD de tests, vous avez fini avec, donc vous voulez la supprimer.

Syntaxe :

```
DROP DATABASE nombdd ;
```

Vous remarquerez que DATABASE est au singulier, car on ne souhaite supprimer qu'une seule DATABASE.

Exemple :

```
DROP DATABASE test ;
```

Gestion de la structure des tables

Visualiser le nom des tables existantes dans la BDD

Contexte : Vous voulez visualiser le nom de toutes les tables existantes dans la BDD que vous utilisez.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

SHOW TABLES ;

```
MariaDB [exemple_bdd]> SHOW TABLES;
+-----+
| Tables_in_exemple_bdd |
+-----+
| acheter                |
| article                |
| client                 |
| identification          |
+-----+
4 rows in set (0.000 sec)
```

Visualiser la structure d'une table existante dans la BDD

Contexte : Vous voulez visualiser la structure d'une table précise existante dans la base de données. Cela vous permet de voir le nom de ses colonnes (colonne Field), leurs types (colonne type !), leurs options (colonnes Null et Extra), si elles sont configurées en tant que clé primaire (PRI dans la colonne Key) ou en tant que clé étrangère (MUL dans la colonne Key).

Syntaxe :

DESCRIBE **nomdelatable** ;

Exemple :

DESCRIBE **identification** ;

```
MariaDB [exemple_bdd]> DESCRIBE identification;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)   | NO   | PRI | NULL    | auto_increment |
| nom   | varchar(20) | YES  |     | NULL    |                |
| prenom | varchar(20) | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

Créer une table avec ses colonnes et leurs caractéristiques

Syntaxe : CREATE TABLE **nomdelatable** (dans les parenthèses se configurent chaque colonne sous cette forme : (**nom_colonne1** type_colonne1 [options], **nom_colonne2** type_colonne2 [options]) ;

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordnatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordnatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordnatrice Filière WEBDESIGN / PPNUM		

TRAME

Le nom de la colonne et le type de la colonne sont des renseignements **obligatoires** pour construire une colonne.

```
CREATE TABLE nomdelatable (id_nomdelatable INT(5) NOT NULL PRIMARY KEY AUTO INCREMENT, nomcolonne2_nomdelatable VARCHAR(30), nomcolonne3_nomdelatable VARCHAR(30));
```

Exemple :

```
CREATE TABLE identification (id_identification INT(5) NOT NULL PRIMARY KEY AUTO INCREMENT, nom_identification VARCHAR(30), prenom_identification VARCHAR(30));
```

Modifier une table (rajouter, supprimer et modifier une colonne)

Rajouter une colonne dans une table

Contexte : Vous voulez une nouvelle information sur votre BDD, par exemple l'âge de vos utilisateurs, il faut donc rajouter une colonne.

Syntaxe :

```
ALTER TABLE nomdelatable ADD COLUMN nomdelanouvellecolonne typedelanouvellecolonne ;
```

Vous voulez altérer la table (= la modifier) => ALTER TABLE nomdelatable, pour y ajouter une colonne => ADD COLUMN configuration d'une colonne. Je rappelle qu'une colonne doit être obligatoirement composée d'un nom et d'un type. Si vous le souhaitez, vous pouvez en plus spécifier des options, comme lorsque vous déclarez les colonnes à la création de la table.

Exemple :

```
ALTER TABLE identification ADD COLUMN age INT ; => Cela créer dans ma table identification une nouvelle colonne nommée age et de type INT (nombre entier).
```

Modifier une colonne dans une table

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Modifier le nom d'une colonne

Contexte : Vous vous êtes trompés sur le nom d'une colonne, vous voulez la renommer. Lors de cette commande, vous allez changer le nom de la table, mais vous allez devoir aussi saisir à nouveau les informations de configuration de votre colonne, à savoir le type et les options s'il y en a.

Syntaxe :

```
ALTER TABLE nomdelatable CHANGE nomcolonne nouveaunomcolonne
typecolonne [options] ;
```

Exemple :

Vous n'avez pas respecté les conventions de nommage pour votre nouvelle colonne age, il vous fait donc la renommer en respectant nomcolonne_nomtable.

```
ALTER TABLE identification CHANGE age age_identification INT ;
```

Modifier le type ou les options d'une colonne

Contexte : Vous vous êtes trompés sur le type d'une colonne ou sur ses options, vous voulez donc pouvoir la modifier. Attention, cette commande ne permet pas de changer le nom de la table. Pour cela, regarder la section juste au-dessus.

Syntaxe :

```
ALTER TABLE nomdelatable MODIFY nomcolonne typecolonne [options] ;
```

Exemple :

Vous voulez rendre la colonne prénom obligatoire à la saisie d'un utilisateur dans votre BDD, vous allez donc rajouter l'option NOT NULL sur la colonne prénom.

```
ALTER TABLE identification MODIFY prénom VARCHAR(20) NOT NULL;
```

Nous venons de rajouter une option, vous pouvez évidemment utiliser cette commande pour changer le type de la colonne, définir ou modifier la taille du champ, rajouter ou supprimer une option ...

Supprimer une colonne dans une table

Contexte : Vous avez une information dans votre BDD qui ne vous est plus nécessaire, vous voulez supprimer la colonne concernée.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Syntaxe :

```
ALTER TABLE nomdelatable DROP COLUMN nomcolonne;
```

Vous voulez modifier la table **nomdelatable** pour y supprimer la colonne nommée **nomcolonne**.

Exemple :

```
ALTER TABLE identification DROP COLUMN age ;
```

Supprimer une table entière

Contexte : Vous avez créé une table qui ne sert plus aujourd'hui, ou bien vous avez trop de modifications à apporter dessus, il vaut donc mieux la supprimer pour mieux la recréer ensuite.

Syntaxe :

```
DROP TABLE nomdelatable ;
```

Exemple :

```
DROP TABLE identification ;
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Gestion des données contenues dans une table

Visualiser les enregistrements présents dans une table

Visualiser tous les enregistrements dans une table avec toutes les colonnes

Contexte : Vous avez réalisé des enregistrements dans votre table, ou vous en avez modifié ... et vous voulez donc vérifier que vous avez bien fait les choses.

Syntaxe :

```
SELECT * FROM nomdelatable ;
```

Vous voulez sélectionner => SELECT toutes les colonnes => * de la table nomdelatable => FROM nomdelatable ;

Exemple :

```
SELECT * FROM identification ;
```

```
MariaDB [exemple_bdd]> SELECT * FROM identification;
+----+-----+-----+
| id | nom   | prenom |
+----+-----+-----+
| 1  | BONIN | Morgane |
| 2  | JEAN  | Pierre  |
| 3  | TRUC  | Bidule  |
| 4  | MACHIN | Chose   |
+----+-----+-----+
4 rows in set (0.000 sec)
```

À noter que s'il n'y a pas d'enregistrement dans votre table, vous aurez le message suivant :

```
MariaDB [exemple_bdd]> SELECT * FROM article;
Empty set (0.000 sec)
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Visualiser certains enregistrements dans une table avec toutes les colonnes

Syntaxe :

```
SELECT * FROM nomdelatable WHERE id IN (x,y,z);
```

Vous voulez sélectionner => SELECT toutes les colonnes => * de la table nomdelatable => FROM nomdelatable pour les lignes respectant la condition id IN (x,y,z) c'est-à-dire à la fois id=x et id=y et id=z.

Exemple :

```
SELECT * FROM identification WHERE id IN (1,3);
```

```
MariaDB [exemple_bdd]> SELECT * FROM identification WHERE id IN (1,3);
+----+-----+-----+
| id | nom   | prenom |
+----+-----+-----+
| 1  | BONIN | Morgane |
| 3  | TRUC  | Bidule  |
+----+-----+-----+
2 rows in set (0.001 sec)
```

Visualiser tous les enregistrements dans une table avec des colonnes spécifiques

Syntaxe :

```
SELECT nomcolonne1, nomcolonne2 FROM nomdelatable;
```

Vous voulez sélectionner => SELECT nomcolonne1, nomcolonne2 => les colonnes nommées nomcolonne1, nomcolonne2 de la table nomdelatable => FROM nomdelatable.

Exemple :

```
SELECT nomcolonne1, nomcolonne2 FROM identification;
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

```
MariaDB [exemple_bdd]> SELECT nom, prenom FROM identification;
+-----+-----+
| nom   | prenom |
+-----+-----+
| BONIN | Morgane |
| JEAN  | Pierre |
| TRUC  | Bidule |
| MACHIN | Chose  |
+-----+-----+
4 rows in set (0.000 sec)
```

À noter, que les colonnes s'affichent dans le sens où vous les avez écrites dans la commande :

```
MariaDB [exemple_bdd]> SELECT prenom,nom FROM identification;
+-----+-----+
| prenom | nom   |
+-----+-----+
| Morgane | BONIN |
| Pierre  | JEAN  |
| Bidule  | TRUC  |
| Chose   | MACHIN |
+-----+-----+
4 rows in set (0.000 sec)
```

Insérer une ligne de données (un enregistrement) dans une table avec toutes les valeurs

Contexte : Cette commande est à utiliser dans le cas où on remplit notre table avec une ligne qui va contenir une valeur par colonne, c'est-à-dire que chaque colonne sera complétée.

Syntaxe : INSERT INTO **nomdelatable** VALUES (On écrit les valeurs pour notre enregistrement dans l'ordre d'existence de nos colonnes. La **valeur1** correspond à la **colonne 1**, la **valeur 2** à la **colonne 2** ...)

```
INSERT INTO nomdelatable VALUES ('valeur 1', 'valeur 2', ...)
```

Exemple :

```
INSERT INTO identification VALUES (NULL, 'Morgane', ...)
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Pour la **valeur 1**, j'ai mis NULL afin que la colonne correspondante qui est **Id_identification** qui correspond à notre clé primaire s'auto incrémente toute seule.

Insérer une ligne de données (un enregistrement) dans une table en spécifiant uniquement les valeurs à remplir

Contexte : Cette commande est à utiliser dans le cas où on remplit notre table avec une ligne qui va contenir uniquement certaines valeurs par colonne. C'est-à-dire que certaines colonnes seront donc sans valeur, à condition bien sûr que ces colonnes ne soient pas obligatoires, c'est-à-dire qu'elles n'ont pas été configurées comme NOT NULL.

Syntaxe : INSERT INTO **nomdelatable** (On liste les colonnes auxquelles on veut ajouter des données : **nomcolonneA**, **nomcolonneB** ...) VALUES (On écrit les valeurs pour notre enregistrement dans l'ordre des colonnes qu'on a déclaré juste avant. La **valeurA** correspond à la **colonne A**, la **valeur B** à la **colonne B** ...)

```
INSERT INTO nomdelatable (nomcolonneA, nomcolonneB) VALUES ('valeurA', 'valeur B', ...)
```

Exemple :

```
INSERT INTO identification (prenom_identification, nom_identification) VALUES ('Jorgane', 'BONIN' ...)
```

Cela permet de remplir uniquement les colonnes que l'on souhaite. Les autres colonnes sont remplies par auto-incrémentation s'il elle est paramétrée, ou bien obtienne la valeur NULL.

À noter que pour la saisie des valeurs :

- si on saisit une chaîne de caractère, il faut la placer obligatoirement entre ' ' ou entre « ».
- si on saisit un chiffre, on l'écrit tout simplement : 12

Insérer plusieurs lignes dans une table en une seule commande

Contexte : C'est la même que les deux commandes précédentes, en fonction de si vous remplissez toutes les colonnes, ou si vous remplissez seulement certaines d'entre elles. La différence ce jour à la fin. C'est-à-dire que vous mettez les informations de votre ligne entre parenthèses, pour saisir une deuxième ligne, il suffit de mettre une virgule après la fin de votre parenthèse puis d'en ouvrir une nouvelle paire.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Syntaxe : INSERT INTO **nomdelatable** VALUES (mes valeurs pour la 1re ligne), (mes valeurs pour la seconde ligne), (mes valeurs pour la 3^{ème} ligne) ;

Exemple :

```
INSERT INTO identification VALUES (NULL, 'Morgane', 'BONIN'), (NULL, 'Tom', 'JERRY'), (NULL, 'Harry', 'POTTER');
```

Modifier une donnée spécifique dans une ligne spécifique

Contexte : Vous vous êtes trompés sur le nom de l'un de vos enregistrements, donc vous devez rectifier.

Syntaxe :

```
UPDATE nomdelatable SET nomcolonne=nouvellevaleur WHERE id=x ;
```

Après SET, vous devez spécifier la colonne que vous voulez modifier et indiquer la nouvelle valeur à mettre à la place, cela écrasera l'ancienne, et pour éviter que cette modification soit faite sur toute votre table, vous spécifiez la ligne concernée et ce grâce à l'égalité : id = N°deligne. De cette façon, seule la ligne concernée par cette condition sera modifiée.

```
UPDATE nomdelatable SET nomcolonne=nouvellevaleur WHERE id=x;
```

Exemple :

```
UPDATE identification SET nom='JERRY' where id=2; => Cela changera le nom en JERRY de la ligne d'id 2. À faire à condition d'avoir regardé avant l'id de la ligne qu'on veut modifier bien sûr !
```

Ou bien

```
UPDATE identification SET nom='JERRY' where prenom='Tom'; => Cela changera le nom en JERRY de la ligne ayant le prénom Tom. Mais si plusieurs lignes ont le prénom Tom, cela les changera le nom en JERRY sur chacune d'entre elles. C'est pour cela que la méthode avec l'id est celle qui est recommandée.
```

Supprimer une donnée spécifique dans une ligne spécifique

Contexte : À condition que la colonne accepte le fait que sa donnée soit NULL, c'est-à-dire qu'elle n'ait pas été configurée en NOT NULL, vous pouvez supprimer une donnée sur une ligne en utilisant la

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Morgane BONIN</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

même commande que pour modifier une donnée, vue juste au-dessus, mais au lieu de mettre une valeur dans le champ, vous mettez NULL.

Syntaxe :

```
UPDATE nomdelatable SET nomcolonne=NULL WHERE id=x;
```

Exemple :

```
UPDATE identification SET nom=NULL where id=3; => Cela supprimera le nom de l'enregistrement d'id 3.
```

Supprimer une ligne d'enregistrement entière

Contexte : Une personne se désinscrit et ne souhaite plus apparaître dans votre BDD, vous devez donc supprimer la ligne qui lui est associée.

Syntaxe :

```
DELETE FROM nomdelatable WHERE id=x ;
```

Exemple :

```
DELETE FROM identification WHERE id=2 ; => Cela va supprimer la ligne qui correspond à id=2 ;
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Éléments complémentaires

Les différentes structures conditionnelles possibles

Dans quelques commandes vues précédemment, il y a des conditions qui suivent le mot WHERE.

Il est conseillé de toujours baser ces conditions sur la colonne id, car c'est la colonne qui est la clé primaire de vos tables, c'est donc la seule qui est assurée de ne pas avoir deux fois la même valeur. Mais si vous le souhaitez, vous pouvez vous baser sur n'importe quelle autre commande en suivant la même syntaxe d'écriture.

Nous avons vu que vous pouviez utiliser l'égalité pour appliquer la commande sur une ligne spécifique : **id=x**, et nous avons aussi vu que vous pouviez spécifier plusieurs lignes spécifiques avec **id IN (x, y, z)**.

Vous avez d'autres syntaxes possibles :

id != x ;	<= Exclure la ligne x en particulier
id < x ;	<= Inférieur à x
id <= x ;	<= Inférieur ou égale à x
id > x ;	<= Supérieur à x
id >= x ;	<= Supérieur ou égale à x
id BETWEEN x AND z ;	<= Compris entre x et z (x et z étant compris dans l'intervalle)

Listes des types de données que vous pouvez utiliser sur les colonnes

À la création des colonnes, vous devez définir obligatoirement un type de donnée pour votre colonne.

À noter que lorsque vous saisissez des données dans un champ, c'est-à-dire que vous donnez une valeur à un champ lors d'un enregistrement, vous devez entourer votre valeur par des ' ' ou des « » s'il s'agit d'une chaîne de caractères. S'il s'agit d'un nombre, vous n'en avez pas besoin.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Les types de colonnes les plus courants sont :

INT : Nombre entier. Vous pouvez spécifier entre parenthèses le nombre de chiffres qui seront affichés en BDD, pour un âge mettez INT(2) ou (3) en fonction ;) Pour un téléphone, mettez INT(10).

FLOAT : Nombre à virgule

VARCHAR : Champs de texte avec taille variable, peut contenir jusqu'à 255 caractères, mais vous pouvez le restreindre en spécifiant la taille entre parenthèses VARCHAR(20) pour une colonne prénom par exemple.

DATE : Date sous la forme AAAA-MM-JJ

DATETIME : Date + heure sous la forme AAAA-MM-JJ HH:MM:SS

Il y en a d'autres souvent utilisé :

CHAR : Champs texte dédiés à tes tailles de texte qui ne varient pas par exemple un code postal si vous le stocké sous forme de chaîne de caractères, dans ce cas-là, spécifier la taille entre parenthèses CHAR(5).

TEXT : Champs de texte pouvant contenir beaucoup plus de caractères.

TIME : Heure sous la forme HH:MM:SS

Il en existe plein d'autres, mais beaucoup moins utilisés.

Ces éléments **doivent** être mis en place à la création de la colonne en même temps que la création de la table correspondante (CREATE TABLE), mais ils peuvent être modifiés en modifiant la colonne (ALTER TABLE ... MODIFY).

Listes des contraintes/options que vous pouvez utiliser sur les colonnes

Sur MariaDB vous pouvez mettre en place des options sur vos colonnes, pour les configurer plus spécifiquement. Ces options s'appellent des contraintes.

Il existe différentes contraintes. Les plus importantes sont :

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Morgane BONIN</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

- **NOT NULL** : empêche d'enregistrer une valeur nulle pour la colonne, c'est-à-dire que cela rend obligatoire le fait de saisir une valeur dans cette colonne
- **PRIMARY KEY** : indiquer si cette colonne est considérée comme clé primaire
- **FOREIGN KEY** : permet d'indiquer que cette colonne fait référence à des valeurs préexistantes dans une colonne d'une autre table. C'est la clé étrangère.

En voici d'autres intéressantes :

- **DEFAULT** : attribuer une valeur par défaut si aucune donnée n'est indiquée pour cette colonne lors de l'ajout d'une ligne dans la table. Cette valeur par défaut est NULL.
- **UNIQUE** : les valeurs de la colonne doivent être unique ou NULL, c'est-à-dire qu'à l'exception du marqueur NULL, il ne doit jamais y avoir plus d'une fois la même valeur (pas de doublon possible)

Ces éléments **peuvent** être mis en place à la création de la colonne en même temps que la création de la table correspondante (CREATE TABLE), ou bien en modifiant la colonne (ALTER TABLE ... MODIFY).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

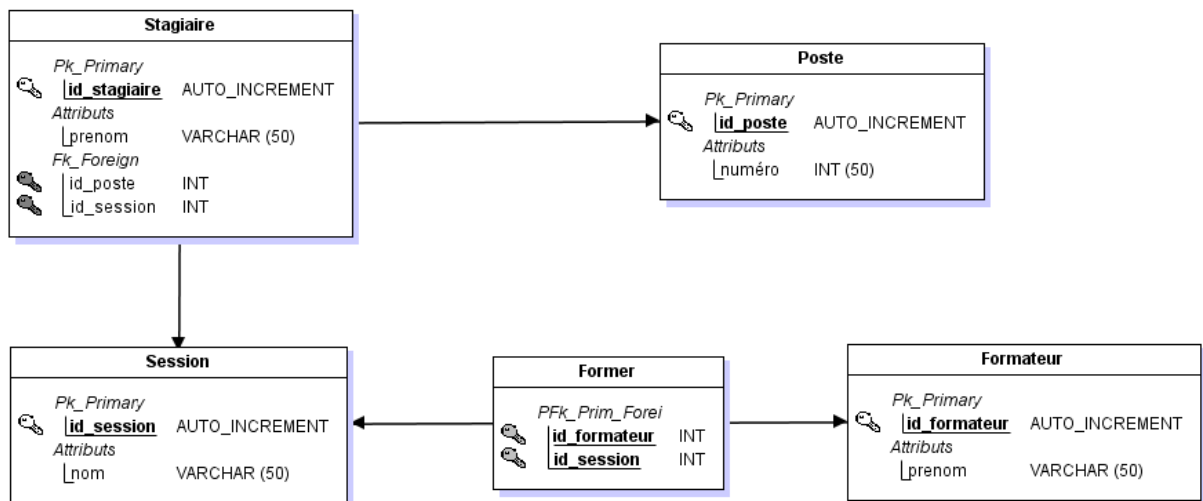
TRAME

Clé primaire

Lors de la conception d'une BDD vous avez découvert le concept de **clé primaire** quand vous avez créé vos MLD. Cet élément est à la base des relations entre nos tables. Il permet d'identifier de manière unique une ligne dans notre BDD.

Nous créons les BDD en nous basant sur les MLD créées.

Prenons l'exemple du cours sur la conception.



Créons la table stagiaire avec les colonnes « id_poste » et « numero » et indiquons que la colonne « id_poste » est la clé primaire de notre table, avec tous les paramètres que cela implique. Par convention et lisibilité ont créé la colonne étant la clé primaire en 1^{er}.

Création table POSTE :

```

MariaDB [exemple_bdd]> CREATE TABLE Poste (
  -> id_poste INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  -> numero INT
  -> );

```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Vous constatez que pour la déclaration de la colonne `id_poste` on a indiqué son type « INT » et ensuite ses autres paramètres (**obligatoires** dans notre cas de clé primaire), qu'on appelle des « contraintes », à savoir :

- NOT NULL : pour indiquer que cette colonne `id_poste` est obligatoire, donc elle ne peut pas être null

- PRIMARY KEY : Évidemment, pour indiquer que c'est cette colonne qui fait office de clé primaire

- AUTO_INCREMENT : pour faire en sorte que la valeur numérique (INT) de notre `id_poste` s'incrémente automatiquement à chaque nouvel enregistrement, de cette façon ce n'est pas à nous de gérer cet id mais au système. Comme cela, on est certains de ne pas avoir deux fois le même id.

Chaque colonne d'id dans chaque table que vous devez créer doit contenir ces contraintes afin de la déclarer correctement comme clé primaire.

```
MariaDB [exemple_bdd]> DESCRIBE Poste;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_poste | int(11) | NO | PRI | NULL | auto_increment |
| numero | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Clé étrangère

Vous avez une table basique « Poste » qui possède deux colonnes, dont une qui est sa clé primaire.

Dans le cadre de notre MLD, cette table est en relation avec Stagiaire par le biais d'une clé étrangère. C'est la combinaison entre la clé primaire et la clé étrangère qui permet de mettre en lien deux tables.

Par convention et lisibilité ont créé les colonnes faisant office de clé étrangère en dernier.

Mise en place d'une clé étrangère à la création de la table

Étant donné que Poste est déjà créé, nous allons créer la table Stagiaire en indiquant la clé étrangère qui fait référence à la table Poste dès la création de la table.

Création Table Stagiaire :

```
MariaDB [exemple_bdd]> CREATE TABLE Stagiaire (
  -> id_stagiaire INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  -> prenom VARCHAR(20),
  -> id_poste INT,
  -> FOREIGN KEY (id_poste) REFERENCES Poste(id_poste)
  -> );
Query OK, 0 rows affected (0.082 sec)
```

On créer nos trois colonnes, id_stagiaire en clé primaire, prenom et id_poste. Car il faut en premier lieu créer la colonne qui va recevoir la relation, ici « id_poste ». Ensuite, on définit cette clé étrangère par la dernière ligne :

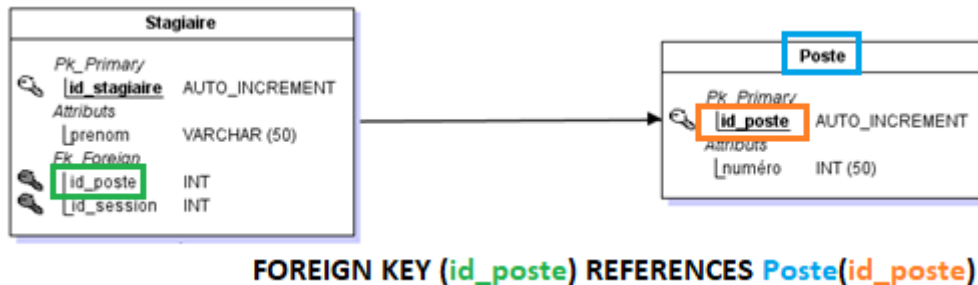
FOREIGN KEY (id_poste) REFERENCES Poste(id_poste)

Qui veut dire qu'on déclare en tant que clé étrangère (Foreign key) la colonne id_poste présente sur la table qu'on est en train de créer, qui fait référence à la clé primaire id_poste de la table Poste.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

En image et en couleur, ça donne :



Pour vérifier que cette clé étrangère à bien été mise en place, faites une description de votre table Stagiaire :

```
MariaDB [exemple_bdd]> DESCRIBE Stagiaire;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id_stagiaire | int(11)   | NO   | PRI | NULL    | auto_increment |
| prenom      | varchar(20) | YES  |     | NULL    |                |
| id_poste     | int(11)   | YES  | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

Vous constatez que la colonne id_poste est déclarée en tant que « MUL » dans la colonne « KEY », MUL = MULTIPLE, cela veut dire que dans notre table la colonne MUL permet d'avoir de multiples références vers d'autre table, autrement dit, il s'agit de notre clé étrangère.

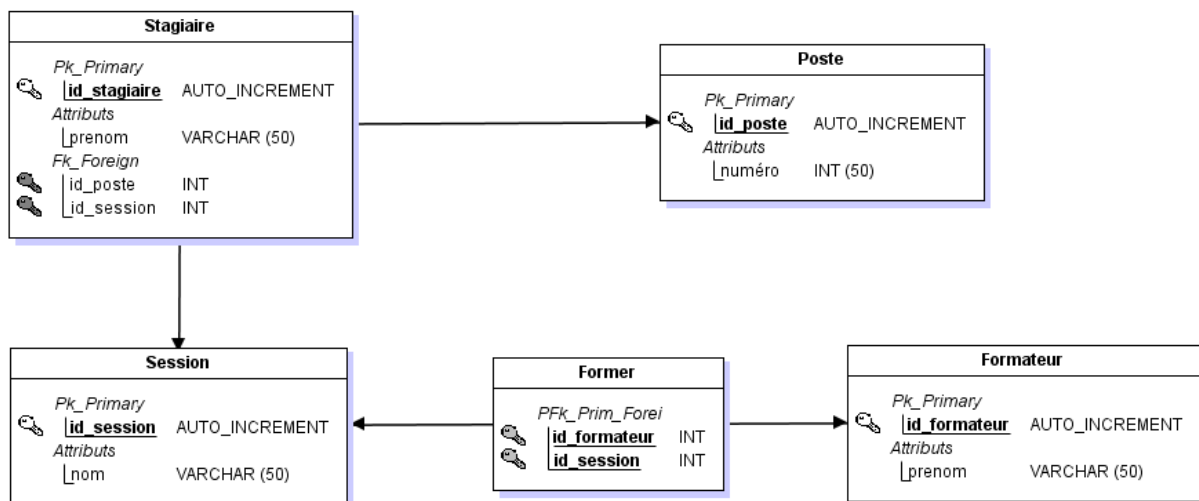
Un second point qui peut être amené à votre réflexion et qui dépend de votre gestion, votre MLD et donc votre vision des choses, il s'agit du fait que la colonne servant de clé étrangère ait le droit d'être NULL ou non. À vous de définir si le champ id_poste est obligé d'être rempli quand on crée un poste. Si la réponse est oui, alors vous devez définir la colonne comme étant NOT NULL. Si la réponse est non, comme dans notre cas, vous pouvez la laisser en NULL.

En effet, lorsqu'on crée un stagiaire, il n'y a pas obligation de lui attribuer un poste (ce à quoi la colonne id_poste correspond).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Mise en place de la clé étrangère avec modification d'une table existante



Reprenons notre MLD et poursuivons sa création.

Nos stagiaires doivent appartenir à une session.

Créons la table Session :

```

MariaDB [exemple_bdd]> CREATE TABLE Session (
  -> id_session INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  -> nom VARCHAR(20)
  -> );
Query OK, 0 rows affected (0.080 sec)
  
```

```

MariaDB [exemple_bdd]> DESCRIBE Session;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id_session | int(11)   | NO   | PRI | NULL    | auto_increment |
| nom        | varchar(20) | YES  |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
  
```

Maintenant que la table Session est créée, on peut modifier notre table Stagiaire pour créer la nouvelle clé étrangère, celle qui fait le lien avec la Session.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Modification table Stagiaire :

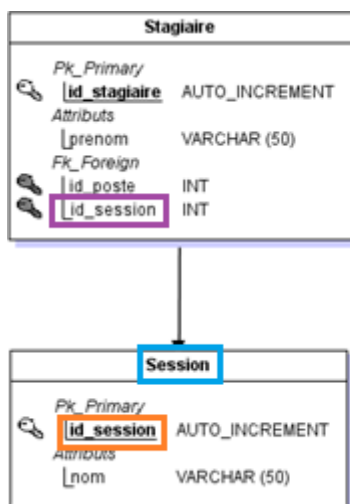
```
MariaDB [exemple_bdd]> ALTER TABLE Stagiaire
-> ADD COLUMN id_session INT,
-> ADD FOREIGN KEY (id_session) REFERENCES Session(id_session);
```

On modifie la table Stagiaire,

=> On ajoute la colonne id_session en type INT

=> On déclare cette nouvelle colonne en tant que clé étrangère qui fait référence à la clé primaire nommée id_session de la table Session.

En image et en couleur, ça donne :



ALTER TABLE Stagiaire

```
-> ADD COLUMN id_session INT,
-> ADD FOREIGN KEY (id_session)
-> REFERENCES Session(id_session);
```

À présent votre table Stagiaire est correctement paramétrée avec ses deux clés étrangères :

```
MariaDB [exemple_bdd]> DESCRIBE Stagiaire;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id_stagiaire | int(11)   | NO   | PRI | NULL    | auto_increment |
| prenom      | varchar(20) | YES  |     | NULL    |              |
| id_poste    | int(11)   | YES  | MUL | NULL    |              |
| id_session  | int(11)   | YES  | MUL | NULL    |              |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Pour le voir de manière plus certaine, connectez-vous à Adminer et allez sur votre table Stagiaire.

Table: Stagiaire

[Afficher les données](#) [Afficher la structure](#) [Modifier la table](#)

Colonne	Type	Commentaire
id_stagiaire	int(11) <i>Incrément automatique</i>	
prenom	varchar(20) <i>NULL</i>	
id_poste	int(11) <i>NULL</i>	
id_session	int(11) <i>NULL</i>	

Clés étrangères

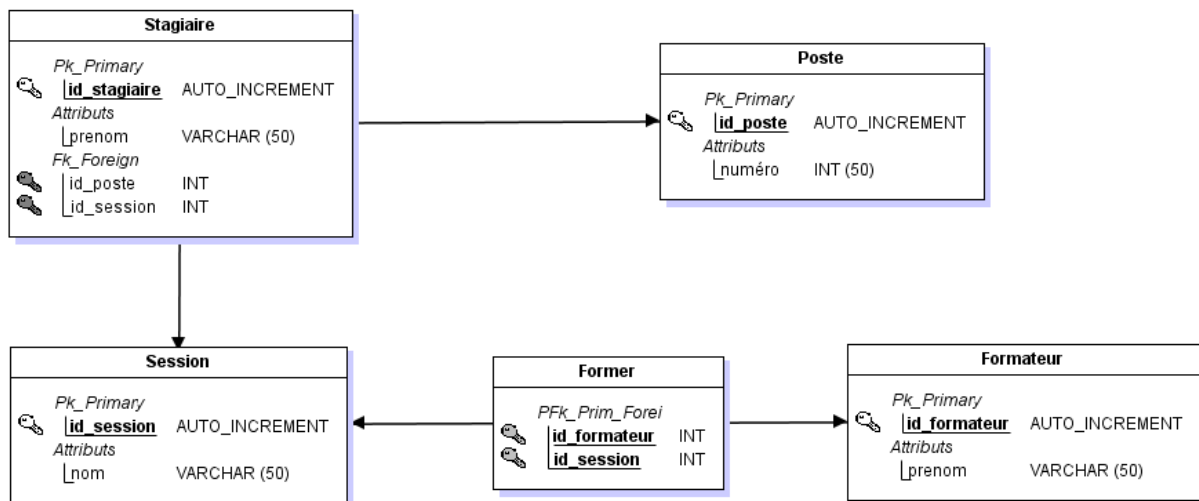
Source	Cible	ON DELETE	ON UPDATE	
id_poste	Poste(<i>id_poste</i>)	RESTRICT	RESTRICT	Modifier
id_session	Session(<i>id_session</i>)	RESTRICT	RESTRICT	Modifier

[Ajouter une clé étrangère](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Table d'association



Nous savons à présent créer et paramétrer une clé étrangère à la création même de la table, mais aussi en l'ajoutant après lors d'une modification.

Attaquons-nous à un cas légèrement plus complexe, à savoir la table d'association.

En premier, créons la table Formateur :

```

MariaDB [exemple_bdd]> CREATE TABLE Formateur (
  -> id_formateur INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  -> prenom VARCHAR(20)
  -> );
Query OK, 0 rows affected (0.058 sec)

MariaDB [exemple_bdd]> DESCRIBE Formateur;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id_formateur | int(11)       | NO   | PRI | NULL    | auto_increment |
| prenom      | varchar(20)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)

```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Maintenant, nous avons nos deux tables pour notre relation, à savoir Session et Formateur. Mais grâce à notre MLD on sait qu'on doit mettre en place une table d'association, car une session peut avoir plusieurs formateurs et un formateur peut avoir plusieurs sessions. Comme on est dans une relation multiple (n étant la cardinalité maximum des deux côtés dans notre MCD), cela donne lieu à la table d'association Former.

Mettons en place cette table d'association en créant ses colonnes, en paramétrant sa clé primaire et ses clés étrangères. La spécificité d'une table d'association, comparer à ce que nous venons de faire, c'est que sa clé primaire est composée des deux clés étrangères.

```
MariaDB [exemple_bdd]> CREATE TABLE Former (
  -> id_session INT NOT NULL,
  -> id_formateur INT NOT NULL,
  -> PRIMARY KEY (id_session, id_formateur),
  -> FOREIGN KEY (id_session) REFERENCES Session(id_session),
  -> FOREIGN KEY (id_formateur) REFERENCES Formateur(id_formateur)
  -> );
Query OK, 0 rows affected (0.099 sec)
```

On créer la table d'association Former

=> on créer la colonne id_session

=> on créer la colonne id_formateur

=> on met en place la clé primaire qui est composée de id_session et de id_formateur

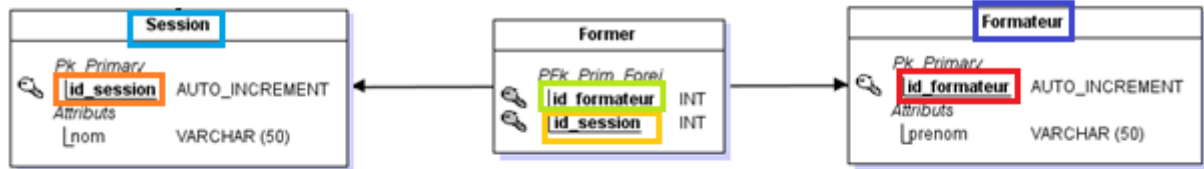
=> on met en place une clé étrangère sur la colonne id_session qui référence la clé primaire id_session de la table Session

=> on met en place la seconde clé étrangère sur la colonne id_formateur qui référence la clé primaire id_formateur de la table Formateur.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

En image et en couleur :



```
CREATE TABLE Former (
-> id_session INT NOT NULL,
-> id_formateur INT NOT NULL,
-> PRIMARY KEY (id_session, id_formateur),
-> FOREIGN KEY (id_session) REFERENCES Session(id_session),
-> FOREIGN KEY (id_formateur) REFERENCES Formateur(id_formateur)
-> );
```

Pour vérifier la bonne mise en place, décrivez votre table :

```
MariaDB [exemple_bdd]> DESCRIBE Former;
+-----+-----+-----+-----+-----+-----+
| Field      | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_session | int(11)| NO   | PRI | NULL    |       |
| id_formateur | int(11)| NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

Vous voyez qu'id_session et id_formateur sont des clés primaires, mais on ne voit pas le fait qu'elles soient clés étrangères. Pour cela, connectez-vous sur Adminer :

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Table: Former

[Afficher les données](#)

[Afficher la structure](#)

[Modifier la table](#)

[Nouvel é](#)

Colonne	Type	Commentaire
id_session	int(11)	
id_formateur	int(11)	

Clés étrangères

Source	Cible	ON DELETE	ON UPDATE	
id_session	Session(id_session)	RESTRICT	RESTRICT	Modifier
id_formateur	Formateur(id_formateur)	RESTRICT	RESTRICT	Modifier

[Ajouter une clé étrangère](#)

On constate bien que les colonnes id_session et id_formateur sont des clés étrangères.

Vous avez donc bien construit votre BDD en suivant votre MLD.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Insertion des enregistrements avec les clés étrangères

À présent, pour finaliser, il faut que vous mettiez des valeurs dans vos tables, c'est-à-dire que vous mettiez en place des enregistrements dans vos tables.

Cela se passe exactement comme avant, c'est-à-dire que les colonnes servant de clés étrangères sont des colonnes comme les autres, elles attendent donc une saisie de valeur. La valeur doit être l'id correspondant à la ligne que vous voulez mettre en lien.

Dans un premier temps, faisons deux enregistrements sur la table Poste, sur la table Session et sur la table Formateur, car ces tables n'ont pas de clés étrangères.

```
MariaDB [exemple_bdd]> INSERT INTO Poste VALUES (NULL, 4),(NULL,5);
Query OK, 2 rows affected (0.015 sec)
Records: 2 Duplicates: 0 Warnings: 0

MariaDB [exemple_bdd]> SELECT * FROM Poste;
+-----+-----+
| id_poste | numero |
+-----+-----+
| 1 | 4 |
| 2 | 5 |
+-----+-----+
2 rows in set (0.000 sec)
```

```
MariaDB [exemple_bdd]> INSERT INTO Session VALUES
-> (NULL, 'TSSR19-03'),
-> (NULL, 'TSSR19-04'),
-> (NULL, 'TSSR20-01');
Query OK, 3 rows affected (0.014 sec)
Records: 3 Duplicates: 0 Warnings: 0

MariaDB [exemple_bdd]> SELECT * FROM Session;
+-----+-----+
| id_session | nom |
+-----+-----+
| 1 | TSSR19-03 |
| 2 | TSSR19-04 |
| 3 | TSSR20-01 |
+-----+-----+
3 rows in set (0.000 sec)
```

```
MariaDB [exemple_bdd]> INSERT INTO Formateur VALUES
-> (NULL, 'Morgane'),
-> (NULL, 'Mathieu'),
-> (NULL, 'Christophe'),
-> (NULL, 'Bastien');
Query OK, 4 rows affected (0.016 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [exemple_bdd]>
MariaDB [exemple_bdd]>
MariaDB [exemple_bdd]> SELECT * FROM Formateur;
+-----+-----+
| id_formateur | prenom |
+-----+-----+
| 1 | Morgane |
| 2 | Mathieu |
| 3 | Christophe |
| 4 | Bastien |
+-----+-----+
4 rows in set (0.000 sec)
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Créons des stagiaires, qui sont dans une session et possèdent peut-être des pc :

```
MariaDB [exemple_bdd]> describe Stagiaire;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id_stagiaire | int(11)   | NO   | PRI | NULL    | auto_increment |
| prenom      | varchar(20) | YES  |     | NULL    |              |
| id_poste    | int(11)   | YES  | MUL | NULL    |              |
| id_session  | int(11)   | YES  | MUL | NULL    |              |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)

MariaDB [exemple_bdd]> INSERT INTO Stagiaire VALUES
-> (NULL, 'Thor', NULL, 2),
-> (NULL, 'Hulk', 1, 3),
-> (NULL, 'Iron', 2, NULL),
-> (NULL, 'Lucky', NULL, NULL),
-> (NULL, 'Captain', 1, 1);
Query OK, 5 rows affected (0.015 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [exemple_bdd]> SELECT * FROM Stagiaire;
+-----+-----+-----+-----+
| id_stagiaire | prenom | id_poste | id_session |
+-----+-----+-----+-----+
| 1 | Thor | NULL | 2 |
| 2 | Hulk | 1 | 3 |
| 3 | Iron | 2 | NULL |
| 4 | Lucky | NULL | NULL |
| 5 | Captain | 1 | 1 |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

Pour vous faciliter le travail, je vous conseille de faire comme moi. En premier, décrivez votre table, cela permet de voir dans quel ordre les colonnes sont déclarées, de cette façon vous pouvez créer plusieurs stagiaires en une seule ligne de commande, sans avoir à déclarer les colonnes avant.

Puis affichez vos saisies.

Moi j'ai 5 stagiaires :

- le 1^{er}, Thor, n'a pas de poste (NULL), et est dans la session d'id 2, je regarde le contenu de la table Session et je vois que celle d'id 2 = TSSR19-04.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

- le 2nd, Hulk, possède un poste, le numéro 4 et appartient à la session d'id 3 = TSSR20-01.

Ainsi de suite, je suis la valeur de mes clés étrangères en suivant leur lien vers les clés primaires associées dans les tables respectives pour voir à qu'elle valeur cela correspond.

Sur Adminer, si vous visualisez la table Stagiaire, vous voyez les données, et les valeurs saisies dans les

Sélectionner: Stagiaire

[Afficher les données](#) [Afficher la structure](#) [Modifier la table](#) [Nouvel élément](#)

SELECT * FROM `Stagiaire` LIMIT 50 (0,000 s) [Modifier](#)

<input type="checkbox"/> Modification	id_stagiaire	prenom	id_poste	id_session
modifier	1	Thor	NULL	2
modifier	2	Hulk	1	3
modifier	3	Iron	2	NULL
modifier	4	Lucky	NULL	NULL
modifier	5	Captain	1	1

[Importer](#)

colonnes servant de clés étrangères sont cliquables, afin de vous amener directement sur la valeur correspondante. Cela permet de vérifier que votre clé étrangère fonctionne.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Testons avec la table d'association !

```
MariaDB [exemple_bdd]> INSERT INTO Former VALUES
-> (1,3),
-> (1,2),
-> (2,2),
-> (2,1),
-> (3,1),
-> (3,4),
-> ;
Query OK, 6 rows affected (0.015 sec)
Records: 6 Duplicates: 0 Warnings: 0

MariaDB [exemple_bdd]> SELECT * FROM Former;
+-----+-----+
| id_session | id_formateur |
+-----+-----+
| 1 | 2 |
| 1 | 3 |
| 2 | 1 |
| 2 | 2 |
| 3 | 1 |
| 3 | 4 |
+-----+-----+
6 rows in set (0.000 sec)
```

Sur Adminer, toutes nos valeurs sont bien cliquables et renvoi donc bien aux données réelles.

Sélectionner: Former

[Afficher les données](#) [Afficher la structure](#) [Modifier la table](#) [Nouvel élément](#)

SELECT * FROM `Former` LIMIT 50 (0,000 s) [Modifier](#)

<input type="checkbox"/> Modification	id_session	id_formateur
<input type="checkbox"/> modifier	1	2
<input type="checkbox"/> modifier	1	3
<input type="checkbox"/> modifier	2	1
<input type="checkbox"/> modifier	2	2
<input type="checkbox"/> modifier	3	1
<input type="checkbox"/> modifier	3	4

[Importer](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Morgane BONIN	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Évidemment, les données vues comme ça, c'est un peu barbare, suivre le lien à la main pour savoir qui est associé à qui. Mais souvent une BDD est utilisée dans le cadre d'un site web, qui va permettre cet affichage simple des informations.

Voici le contenu d'un site internet basique reprenant les infos que nous avons mis dans notre BDD.

Vous verrez le contenu de chaque table, mais aussi ce qu'on peut faire grâce à cela, c'est-à-dire obtenir une table avec directement le nom des éléments liées et non pas leurs ID. C'est ce qui nous intéresse.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Morgane BONIN</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE	29/05/2019	29/04/2020
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI : Coordinatrice Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

TRAME

Affichage basique du contenu d'une BDD sur un site

Contenu de ma table Poste

Ce qui existe

Contenu de la table Poste tel que vous pouvez le voir en faisant la commande SELECT * FROM Poste.

Id	Numéro de poste
1	4
2	5

Contenu de ma table Session

Ce qui existe

Contenu de la table Session tel que vous pouvez le voir en faisant la commande SELECT * FROM Session.

Id	Nom
1	TSSR19-03
2	TSSR19-04
3	TSSR20-01

Contenu de ma table Formateur

Ce qui existe

Contenu de la table Formateur tel que vous pouvez le voir en faisant la commande SELECT * FROM Formateur.

Id	Prénom
1	Morgane
2	Mathieu
3	Christophe
4	Bastien

Contenu de ma table Stagiaire

Ce qui existe

Contenu de la table Stagiaire tel que vous pouvez le voir en faisant la commande SELECT * FROM Stagiaire. Dans le cadre des 3 tables précédentes, cet affichage ne nous posait pas de problème, car nous avions directement toutes les informations nécessaires à notre compréhension. Or, pour les tables Stagiaire et Former, c'est un peu embêtant de devoir suivre la relation entre les ID pour savoir quel stagiaire appartient à quelle session ou quel formateur forme quelle session. Ce contenu "simple" c'est ce que vous voyez juste en dessous.

Id	Prénom	Id_poste	Id_session
1	Thor	1	2
2	Hulk	1	3
3	Iron	2	
4	Lucky		
5	Captain	1	1

Ce qui nous interesse

Dans le cadre d'un site web par exemple, ce qui nous interesse, c'est ce qu'on peut faire avec ces données. La première chose étant d'afficher la valeur réelle de nos id en relation. Comme on peut le voir sur le tableau ci-dessous nous visualisons le nom de nos stagiaires, le numéro de leur pc s'ils en ont un, et le nom de leur session.

Ce tableau, c'est ce qui nous permet d'utiliser une BDD pour structurer nos données, les relier entre elles, éviter les doublons, tout en pouvant les utiliser sur notre site directement afin d'avoir quelque chose de logique et de compréhensible.

Id	Prénom	Id_poste	Id_session
1	Thor		TSSR19-04
2	Hulk	4	TSSR20-01
3	Iron	5	
4	Lucky		
5	Captain	4	TSSR19-03

Contenu de ma table Former

Ce qui existe

Contenu de la table Former tel que vous pouvez le voir en faisant la commande SELECT * FROM Former.

Id_session	id_formateur
1	2
1	3
2	1
2	2
3	1
3	4

Ce qui nous interesse

Contenu de la table Former avec les noms des sessions et des formateurs à la place de ID. C'est cette information là qu'on utilisera sur notre site.

Id_session	id_formateur
TSSR19-03	Mathieu
TSSR19-03	Christophe
TSSR19-04	Morgane
TSSR19-04	Mathieu
TSSR20-01	Morgane
TSSR20-01	Bastien

Aute

IAJ :

Morgane BONIN

Responsable Secteur NUMERIQUE OCCITANIE

29/05/2019

29/04/2020

Florence CALMETTES :
Coordinatrice Filière SYST. & RESEAUX

Marc CECCALDI :
Coordinatrice Filière DEVELOPPEMENT

Sophie POULAKOS :
Coordinatrice Filière WEBDESIGN / PPNUM

Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.