

## TP 2 MySQL : import et export de données, requête avec jointure interne

Un des buts de ce TP est d'utiliser les outils d'importation et d'exportation.

## 1 Créer et exécuter un script de création de tables

Écrire dans un fichier script.sql le script de création des tables correspondant au modèle suivant :

**ARTICLE**(idArticle, designation, prix)  
**COMMANDE**(idCommande, dateCommande, #idClient)  
**LIGNE**(#idCommande, #idArticle, quantite)  
**CLIENT**(idClient, nom, ville)

Pour exécuter ce script :

```
mysql> source script.sql ;
```

- Supprimer toutes les tables si elles existent, créer toutes les tables si elles n'existent pas.

Utiliser les mots clés **IF EXISTS** lors de la suppression de la table et **IF NOT EXISTS** lors de la création des tables.

## 2 Importer des données

L'objectif est d'importer des données stockées dans un fichier texte.

Utiliser les données du fichier excel en enregistrant chaque feuille au format csv (séparateur de champ : , et séparateur de texte : vide )

Utiliser le nom de la table comme nom de Fichier

Exemple de format pour un fichier : **CLIENT.csv**

```
, Dupont, Lyon
, Pierre, Paris
, Durand, Lille
```

Procéder de la même manière avec les autres tables (onglets dans le tableur) pour créer les autres fichiers de données.

**REMARQUE** : La première virgule de la ligne de l'exemple signifie que le premier champ est NULL ( **AUTO\_INCREMENT** ). Faire attention aux clés étrangères et au format de date : yyyy/mm/dd ou yyyy-mm-dd .

Lancer la commande ci dessous (**dans votre script SQL** ) :

```
mysql> LOAD DATA LOCAL INFILE 'CLIENT.csv' INTO TABLE CLIENT FIELDS TERMINATED BY ',';
```

Faire de même avec les autre fichiers.

**ATTENTION** à l'ordre d'importation des données, c'est dans le même ordre généralement que la création des tables

**ATTENTION** problème résolu : si l'importation des données ne fonctionne pas sur **DataGrip** même en indiquant le chemin absolu du fichier ; configurer datagrip ou utiliser **un terminal classique pour exécuter votre script et importer les données**

## 2.1 problème d'encodage de caractères dans les chaînes de caractères

voir l'encodage de la table

```
SHOW CREATE TABLE ARTICLE;
```

ajouter un encodage au caractère dans la table

```
CREATE TABLE IF NOT EXISTS ARTICLE (
....
)CHARACTER SET 'utf8' ;
```

le terminal mysql n'est pas forcément en `utf8` et l'import des données n'est donc pas forcément en `utf8`

forcer l'encodage des caractères lors de l'importation des données

```
LOAD DATA LOCAL INFILE 'CLIENT.csv' INTO TABLE CLIENT CHARACTER SET utf8
FIELDS TERMINATED BY ',';
```

## 2.2 problème pour utiliser la commande "load" sur MySQL (oracle)

ATTENTION parfois problème sur mysql(oracle), se connecter avec l'option `--local_infile` :

sur [stackoverflow](#)

```
mysql>SHOW GLOBAL VARIABLES LIKE 'local_infile';
```

- Se connecter avec l'option `--local_infile` :

`mysql --user=[Utilisateur] --password=[MotDePasse] --host=localhost --database=[baseDeDonnées] --local_infile` souvent ne fonctionne pas

- `SET GLOBAL local_infile=ON` – souvent ne fonctionne pas
- configurer mysql

éditer le fichier de configuration de mysql

```
sudo nano /etc/mysql/my.cnf
```

ajouter les lignes suivantes à la fin :

```
[mysqld]
local_infile=ON
```

redémarrer le service mysql ( `service --status-all` )

```
sudo service mysql restart
```

sur une [version d'ubuntu](#), il fallait ajouter dans le fichier `/etc/mysql/my.cnf`

```
[mysql]
local_infile=ON
```

sur windows, éditer le fichier `C:\tools\mysql\current\my.ini` (installation avec *chocolatey*)

## 3 Exporter des données : mysqldump

Pour exporter une base (depuis un terminal « shell ») dans le terminal :

```
mysqldump --host=serveurmysql --user=votrelogin --password=motdePasse votreBaseDeDonnee
```

Pour exporter une base (depuis un terminal « shell ») dans un fichier :

```
mysqldump --host=serveurmysql --user=votrelogin --password=motdePasse votreBaseDeDonnee > base.sql
```

ATTENTION depuis peu la commande fonctionne, mais elle affiche un message d'erreur : [stackexchange](#) : ajouter `--no-tablespaces`

```
mysqldump --host=serveurmysql --user=votrelogin --password=motdePasse --no-tablespaces votreBaseDeDonnee > base.sql
```

On obtient un script de création très complet (base.sql) + les INSERT de données. On peut donc l'exécuter de la même façon qu'en partie 1.

Pour n'exporter que certaines tables, rajouter le nom des tables derrière la base de données.

```
mysqldump --host=host --user=votrelogin --password=motdePasse votreBaseDeDonnee t1 t2 > base.sql
```

## 3.1 sauvegarder la base de données dans un fichier avec la date d'aujourd'hui

faire un backup sur un [SE linux](#)

```
echo nom_fichier_backup_$(date +%Y-%m-%d-%H.%M.%S).sql

mysqldump --host=host --user=votrelogin --password=motdePasse votreBaseDeDonnee > nom_fichier_backup_$(date +%Y-%m-%d-%H.%M.%S).sql
```

Récupérer la date système sur [Windows](#), c'est un peu plus compliqué

Exemple final :

```
amillet@amillet-pc:~$ mysqldump --user=votreLogin --password=votreMotDePasse --host=localhost --no-tablespaces
BDD votreLogin LIGNE COMMANDE ARTICLE CLIENT > backup_tp2_$(date +%Y-%m-%d-%H.%M.%S).sql
amillet@amillet-pc:~$ ls backup_tp2*
backup_tp2 2022-01-19-08 45 49.sql
```

depuis un terminal

```
MariaDB [BDD votreLogin]> system mysqldump --user=votreLogin --password=votreMotDePasse --host=localhost
--no-tablespaces BDD_votreLogin LIGNE COMMANDE ARTICLE CLIENT > backup_tp2_$(date +%Y-%m-%d-%H.%M.%S).sql
MariaDB [BDD votreLogin]> system ls back*
backup_tp2 2022-01-19-08 45 49.sql backup_tp2 2022-01-19-08 48 49.sql backup_tp2 2022-01-19-08 49 36.sql
```

depuis un terminal connecté à mysql avec la commande **system**

## 3.2 planification de tâches (sauvegarde de la base de données)

sur **linux**

- “exemple tâche cron”
- “exemple 2 tâche cron”
- “exemple 3 tâche cron”
- “tuto crontab”

sur **windows**

“schtask sur windows”

**INFO** : pour ajouter un **terminal dans DataGrip**

- Dans le menu : [File > Setting] ; choisir [Plugins] Sélectionner “Terminal”
- Une fois installer; Pour configurer le chemin par défaut dans le terminal
  - Dans le menu : [File > Setting] ; choisir [Tools] Sélectionner “Terminal” et configurer “Start Directory”

<https://www.jetbrains.com/help/datagrip/terminal-emulator.html>

## 4 Modifier la structure des tables

- Si on supprime le client 'Mutz', que se passe t'il ?

On désire modifier les contraintes pour pouvoir supprimer en cascade les commandes d'un client (lignes de commandes et commandes)

Modifier les contraintes sans recréer vos tables (**ALTER TABLE ....**). Utiliser les mots clés « **ON DELETE CASCADE** ».

Vérifier les contraintes en regardant la structure de la table.

Faire un essai et vérifier que le contenu des 3 tables a été modifié.

- Exemple de script

```
ALTER TABLE COMMANDE DROP FOREIGN KEY fk_commande_client;
ALTER TABLE LIGNE DROP FOREIGN KEY fk_ligne_commande;
ALTER TABLE LIGNE DROP FOREIGN KEY fk_ligne_article;

ALTER TABLE COMMANDE ADD CONSTRAINT fk_commande_client FOREIGN KEY (idClient)
REFERENCES CLIENT(idClient) ON DELETE CASCADE;
ALTER TABLE LIGNE ADD CONSTRAINT fk_ligne_commande FOREIGN KEY (idCommande)
REFERENCES COMMANDE(idCommande) ON DELETE CASCADE;
ALTER TABLE LIGNE ADD CONSTRAINT fk_ligne_article FOREIGN KEY (idArticle)
REFERENCES ARTICLE(idArticle) ON DELETE CASCADE;

SHOW CREATE table COMMANDE;
SHOW CREATE table LIGNE;

DELETE FROM CLIENT WHERE nom LIKE "Mutz";
```

- Afficher le contenu des tables, conclusion
- **Mettre en commentaire ce script pour la suite**

## 5 Quelques requêtes

- **R1** : Liste des clients dont la ville est Belfort et dont le nom commence par la lettre « m », « e » ou « d » (majuscule et minuscule). Le résultat est ordonné par nom.

```
+-----+
| nom   |
+-----+
| DECOCK |
| duguet |
| ESSENBURGER |
| METTEY |
| molin  |
+-----+
```

- **R2** : Liste des articles dont le prix est compris entre 6 et 10 Euros et dont la désignation possède le chaîne de caractères « lég ». Le résultat est ordonné par désignation.

```
+-----+-----+
| designation | prix |
+-----+-----+
| Haricot ( légume ) | 6.50 |
| Petit pois ( légume ) | 6.00 |
+-----+-----+
```

**RAPPEL COURS : Les jointures internes. Les enregistrements sont retournés quand la condition est vraie dans les 2 tables. C'est l'une des jointures les plus communes.**

commencer par tester les requêtes suivantes :

```
SELECT * FROM COMMANDE ;

SELECT * FROM CLIENT ;

SELECT * FROM CLIENT, COMMANDE ;
```

conclusion

Il existe 2 manières de faire une **jointure interne** :

```
SELECT * -- *Ancienne syntaxe*
FROM CLIENT, COMMANDE
WHERE CLIENT.idClient = COMMANDE.idClient;

SELECT *
FROM CLIENT -- *Syntaxe actuelle*
INNER JOIN COMMANDE
ON CLIENT.idClient = COMMANDE.idClient;
```

Remarques :

- \* le mot clé « INNER » est optionnel
- \* les contraintes sur les tables entre clé primaire et clé étrangère ne lient pas les tables lorsque l'on fait des requêtes

autres exemples

```

)SELECT CLIENT.nom, COMMANDE.dateCommande
FROM CLIENT, COMMANDE
WHERE
CLIENT.idClient = COMMANDE.idClient
AND CLIENT.nom LIKE "Mutz"
)ORDER BY COMMANDE.dateCommande ;

)SELECT cl.nom, co.dateCommande
FROM CLIENT AS cl, COMMANDE AS co
WHERE
cl.idClient = co.idClient
AND cl.nom LIKE "Mutz"
)ORDER BY co.dateCommande ;

)SELECT CLIENT.nom, COMMANDE.dateCommande
FROM CLIENT
JOIN COMMANDE ON CLIENT.idClient = COMMANDE.idClient
WHERE
-- INNER JOIN
CLIENT.nom LIKE "Mutz"
)ORDER BY COMMANDE.dateCommande;

)SELECT cl.nom, co.dateCommande
FROM CLIENT AS cl
JOIN COMMANDE AS co ON cl.idClient = co.idClient
WHERE
-- INNER JOIN
cl.nom LIKE "Mutz"
)ORDER BY co.dateCommande;

```

- R3 : première jointure : Dates de toutes les commandes du client de nom 'Mutz' ordonnées par date

nom	dateCommande
Mutz	2020-09-19
Mutz	2020-12-09

- R4 : Faire une requête pour afficher les articles (prix et quantité commandée) des commandes du client de nom 'Mutz' :

nom du client, désignation du produit, prix du produit, quantité commandée, numéro de commande

nom	désignation	prix	quantite	idCommande
Mutz	gâteau fraise (pâtisserie)	5.25	27	4
Mutz	Tomate ( légume )	1.50	5	4
Mutz	Potiron ( légume )	3.00	5	4
Mutz	Pomme ( fruit )	3.00	5	4
Mutz	Petit pois ( légume )	6.00	6	8

- R5 : Faire une requête pour afficher les articles et le C.A. dans chaque commande) des commandes du client de nom 'Mutz' :

nom du client, désignation du produit, identifiant de la commande et **prix total pour chaque produit : prix du produit \* quantité commandée**. L'affichage est trié par le prix\_total dans l'ordre décroissant.

nom	désignation	idCommande	prix_total
Mutz	gâteau fraise (pâtisserie)	4	141.75
Mutz	Petit pois ( légume )	8	36.00
Mutz	Potiron ( légume )	4	15.00
Mutz	Pomme ( fruit )	4	15.00
Mutz	Tomate ( légume )	4	7.50

- R6 : Faire une requête pour afficher les commandes du client de nom 'Mutz' :

nom du client, numéro de commande , prix total pour chaque commande : somme des (prix du produit \* quantité commandée par produit)

nom	idCommande	prix_total
Mutz	4	179.25
Mutz	8	36.00

- R7 : Afficher le nom des clients ayant passé une commande, l'identifiant de leur commande, le prix total HT, la tva et le prix ttc ordonnées par prix (le prix dans la table est HT et la tva est de 20 %)

nom	idCommande	prix_total_HT	TVA	prix_total_TTC
Rato	6	4.50	0.90	5.40

Walter	7	7.50	1.50	9.00
Goerig	5	12.00	2.40	14.40
Walter	3	35.25	7.05	42.30
Mutz	8	36.00	7.20	43.20
Goerig	1	68.24	13.65	81.89
Rato	2	75.24	15.05	90.29
Mutz	4	179.25	35.85	215.10

**Pour les plus rapides :**

- R8 : affiche pour chaque produit, la quantité commandée, l'année de la commande et l'identifiant de la commande

designation	QteCommande	anneeCommande	idCommande
Choux ( légume )	6	2020	2
Coca cola ( boisson )	12	2020	1
Fenouil ( légume )	4	2019	5
Fenouil ( légume )	4	2020	1
Fenouil ( légume )	3	2021	3
gâteau fraise (pâtisserie)	27	2020	4
Gâteaux chocolat ( pâtisserie)	3	2021	3
Haricot ( légume )	4	2020	1
Mouchoir ( hygiène)	6	2020	2
Perrier ( 1L eau )	NULL	NULL	NULL
Petit pois ( légume )	6	2020	2
Petit pois ( légume )	6	2020	8
Pissenlit ( légume )	NULL	NULL	NULL
Poireau ( légume )	3	2020	6
Poireau ( légume )	3	2021	3
Poivron ( légume )	4	2020	1
Pomme ( fruit )	5	2020	4
Pomme de terre ( légume )	6	2020	2
Potiron ( légume )	5	2020	4
Radis ( légume )	3	2021	3
Salade ( légume )	NULL	NULL	NULL
Tomate ( légume )	5	2020	4
Tomate ( légume )	5	2020	7
Vittel ( 1L eau )	NULL	NULL	NULL

- R9 : afficher les produits, la quantité commandée, l'année de la commande et l'identifiant de la commande des produits commandés en 2020 qui sont dans la liste des produits commandés en 2021

designation	QteCommande	anneeCommande	idCommande
Fenouil ( légume )	4	2020	1
Poireau ( légume )	3	2020	6

- R10 : afficher le nombre de commandes de chaque client en 2020

nom	nbCommande	anneeCommande
Rato	3	2020
Mutz	2	2020
Walter	1	2020
Goerig	1	2020
AMGHAR	0	pas de commande
dirand	0	pas de commande
SAINT DIZIER	0	pas de commande
WOLF	0	pas de commande
TYRODE	0	pas de commande
DECOCK	0	pas de commande
duguet	0	pas de commande
molin	0	pas de commande
COTTARD	0	pas de commande
PETIT	0	pas de commande
METTEY	0	pas de commande
TERRAT	0	pas de commande
chaillet	0	pas de commande
DAROSEY	0	pas de commande
JAOUEN	0	pas de commande
BOUCHAUD	0	pas de commande
PAUTOT	0	pas de commande
LAMOTTE	0	pas de commande
SALVI	0	pas de commande
BISMUTH	0	pas de commande
ALANKAYA	0	pas de commande
ESSENBURGER	0	pas de commande

## 6 Modèle Conceptuel de données

## 7 ERD (Entity Relationship Diagram) pour les plus rapides

Dessiner l'ERD de ce schéma (feuille de papier)

En utilisant le modèle logique de données sur le logiciel **JMerise** et ajouter les relations ManyToOne ( Texte « ManyToOne » et symboles)

Dessiner l'ERD avec les contraintes de l'ORM doctrine :

- chaque entité possède pour clé primaire « **id** » (entier de type « AUTO\_INCREMENT »)

voir ANNEXE sur les « ERD » (logiciel utilisé : <https://www.draw.io/?libs=general;uml> (dans basic, « Entity Relationship Diagram ») )

ANNEXE : faire attention aux points suivants :

**Lors de la création des tables :**

\* Faire attention à l'ordre, on ne crée pas une table avec une contrainte sur une table qui n'existe pas.

\* Si un champ utilise l'attribut « AUTO\_INCREMENT », celui-ci doit être une clé primaire.

**Lors de la suppression des tables :**

Faire attention à l'ordre, on ne supprime pas une table qui a sa clé primaire utilisée dans une contrainte d'une autre table (clé étrangère dans une autre table).

**Lors de la création des contraintes** sur les clés étrangères, en cas d'erreur :

\* Vérifier que vous avez créé le champ de la clé étrangère.

\* Le type des 2 champs reliés par la contraintes doivent être identiques (erreur fréquente : « int » relié à « unsigned int »).

\* ATTENTION aux noms des contraintes : pas de doublons dans la base de données (différents d'un nom de table ....), exemple « fk\_table1\_table2 ».

\* Vérifier que le le nom de l'autre table ainsi que sa clé primaire sont corrects.

**Lors de l'importation des données :**

\* Certaines tables utilisent des clés primaires de type « AUTO\_INCREMENT » : si les champs sont nuls lors de l'importation, la clé primaire doit utiliser l'attribut « AUTO\_INCREMENT ».

\* L'ordre des champs importés est important : **respecter l'ordre de création du modèle logique** (tables).

## 8 ANNEXES

documentation

[https://www.w3schools.com/sql/sql\\_join.asp](https://www.w3schools.com/sql/sql_join.asp)

<http://sql.sh/cours/jointures>

avec le formalisme du framework *laravel*, le schéma aurait la forme suivante :

articles(id, designation, prix)

commandes(id, dateCommande, *client\_id*\*)

lignes(*commande\_id*\*, *article\_id*\*, quantite)

clients(id, nom, ville)

### 8.1 ANNEXE

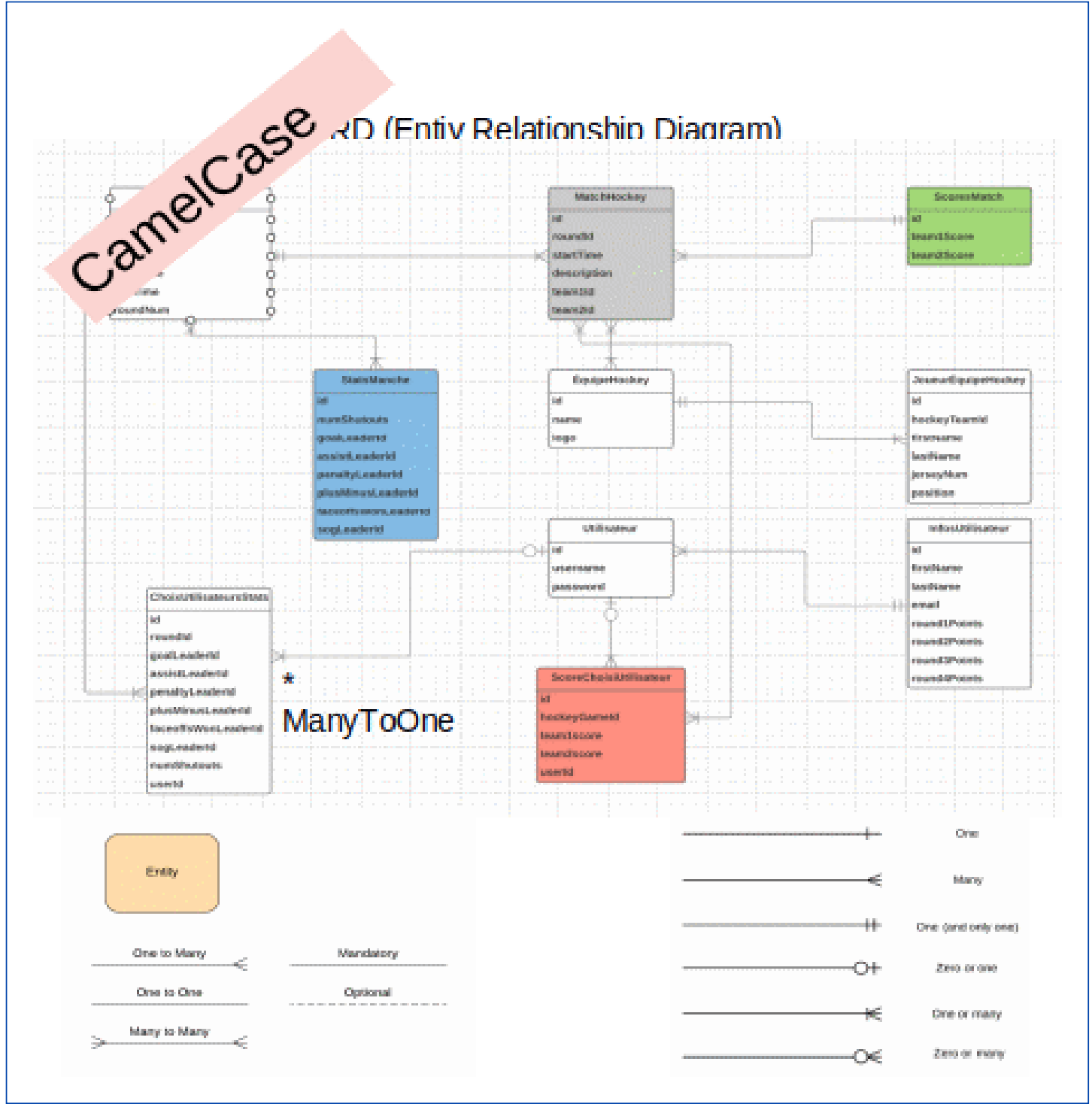
#### 8.1.1 ERD (Entity Relationship Diagram):

Une entité c'est quoi ? => [https://www.lucidchart.com/pages/fr/diagramme-entite-association#section\\_3](https://www.lucidchart.com/pages/fr/diagramme-entite-association#section_3)

<https://www.lucidchart.com/pages/database-diagram/database-models> (ERD)

**ATTENTION: Il faut utiliser du CamelCase**

- [https://www.youtube.com/watch?v=tI\\_5zCLOYug](https://www.youtube.com/watch?v=tI_5zCLOYug)
- <https://stph.scenari-community.org/bdd/gen1/co/pri1c24.html>



ARTICLE

idArticle  
designation  
prix

LIGNE

quantite

COMMANDE

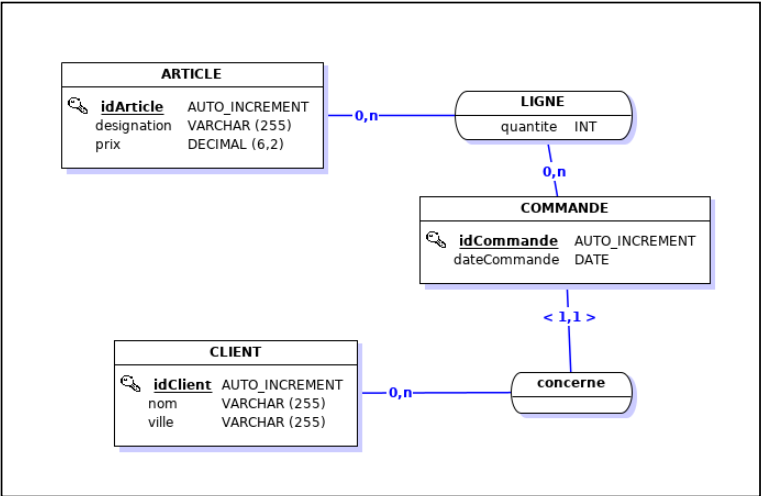
idCommande  
dateCommande

CLIENT

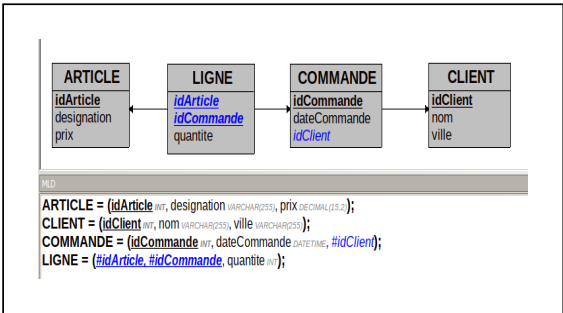
idClient  
nom  
ville

appartient

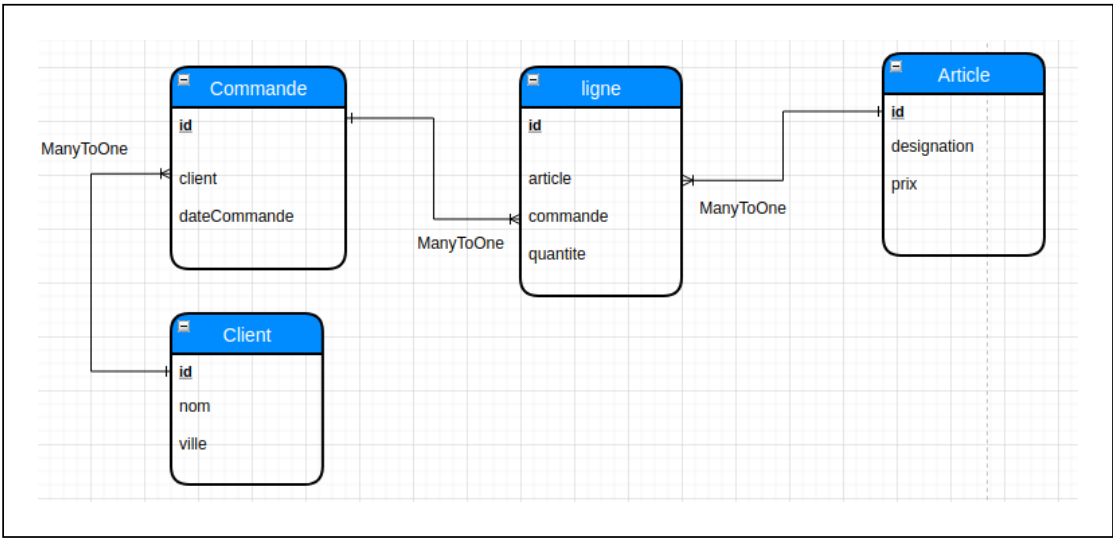
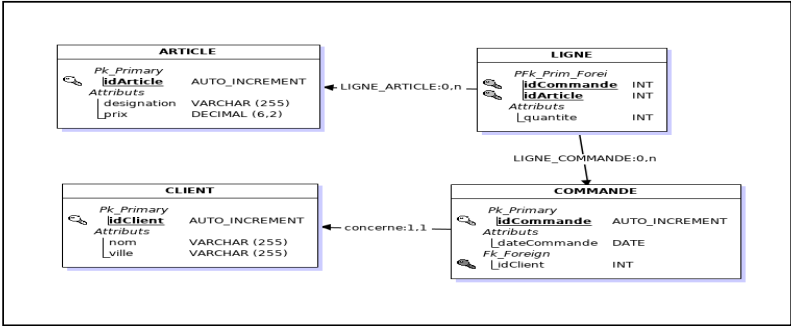




MCD  
(modèle conceptuel de données)  
(schéma à réaliser lors du partiel)



MLD de Jmerise ou Looping



E.R.D. sur Symfony  
( Entity Relationship Diagram )

annexe :  
ANNEXE