

# Export et Import

---

Il est important de faire des sauvegardes de nos BDD. Pour éviter le risque de perte de donnée, suite à un accident ou lorsque l'on travail dessus.

*Si vous travaillez sur un conteneur docker, il faudra penser à transférer le fichier exporté/importé depuis ou vers l'ordinateur*

## EXPORT

Il est possible d'exporter une bdd en un fichier ".sql", Pour cela il faut être déconnecté puis lancer la commande :

```
mysqldump -u nomUtilisateur -p nomBDD > nomFichier.sql  
# Ou  
mariadb-dump -u nomUtilisateur -p nomBDD > nomFichier.sql
```

Il vous demandera votre mot de passe, puis après un temps d'attente plus ou moins long selon la taille de la bdd. créera un fichier là où vous vous trouvez avec votre terminal. Cela dit, cette solution pourra entrainer un problème selon où est ce qu'elle est effectué, effectivement "**powershell**" qui est utilisé comme terminal sur "**vscode**" enregistre les fichiers en "**UTF-16**", or par défaut "**mysql**" attend du "**UTF-8**", on préférera alors :

```
mysqldump -u nomUtilisateur -p nomBDD -r nomFichier.sql
```

("-r" aura aussi retirer des sauts à la ligne ajoutés par windows.) Il est possible d'ajouter des options comme :

```
mysqldump --add-drop-table -u nomUtilisateur -p nomBDD -r nomFichier.sql
```

"--add-drop-table" indique que lors de l'importation, on souhaite que si une table du même nom existe, elle soit supprimé.

Certaines options peuvent être activé par défaut, on peut les vérifier avec :

```
mysqldump --help
```

## IMPORT

Pour importer une bdd il faudra utiliser la commande suivante :

```
mysql -u nomUtilisateur -p nomBDD < nomFichier.sql  
# ou  
mariadb -u nomUtilisateur -p nomBDD < nomFichier.sql
```

ATTENTION ! "<" ne sera pas accepté sur powershell.

L'alternative à cela est de se connecter à notre bdd et lancer la commande suivante après s'être connecté à la BDD :

```
source nomFichier.sql
```