Sécurité des BDD

Sécurité des Systèmes d’Information par Baptiste HAMON

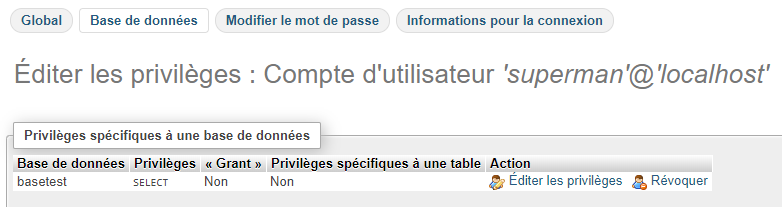
JEROME SARRASIN | IDIR MAKHLOUF

ECE PARIS LYON  - 2020/2021

# Introduction

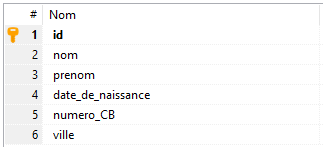
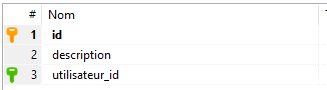
S’inscrivant dans la suite du cours «  Séucrité des bases de données », ce travail vient assoir nos connaissances à travers la pratique en nous permettant de voir les différentes options permettant de sécuriser une base de données(Mysql).

## Création d’utilisateurs et gestion des droits



## Création de BDD et relations entre les tables



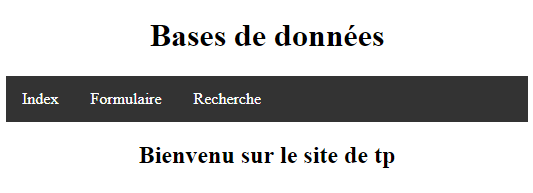


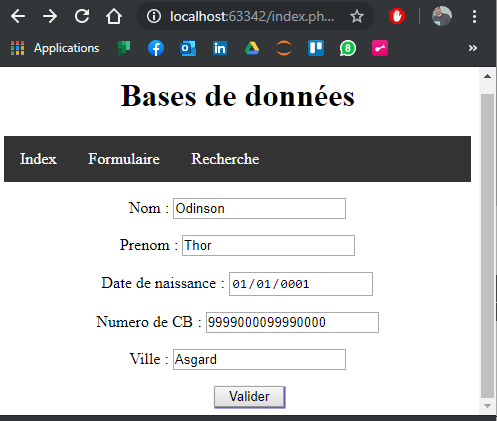
Il y a quatres comportements possibles, lors de la supression ou de la mise à jour d’une clef étrangère : RESTRICT, NO ACTION, SET NULL  et CASCADE.  
  
Ils sont implémanter par les options «  On delete » et « On update »

RESTRICT : est le comportement par défaut. Si l'on essaye de supprimer une valeur référencée par une clé étrangère, l'action est avortée et on obtient une erreur.   
NO ACTION : a exactement le même effet.  
SET NULL : NULL est substitué aux valeurs dont la référence est supprimée ou modifiée.  
CASCADE : supprime/modifie les lignes qui référençaient la valeur supprimée/modifiée

Lors de ce TP nous avons configuré notre cle étrangere NO ACTION pour la supression et CASCADE pour la mise à jour.

## Création de vue et test d’accès aux données



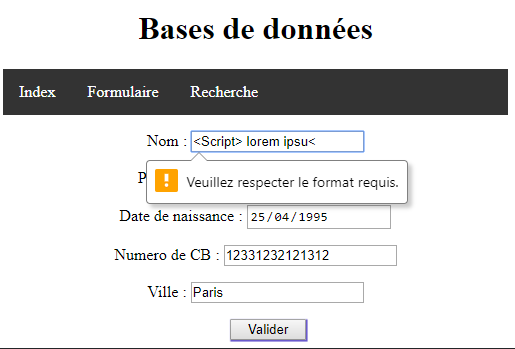


If Valid input

If Non-Valid input

2nd layer of security - input validation on form

2nd layer of security - input validation on form



## Avec droit :

## Sans :

## 

## 

Lien GitHub : <https://github.com/ileets/tp_secu_php>

## Activation de la journalisation

log\_file = C:/MySQL/Logs/query.log

SET GLOBAL general\_log = 'ON';

## Sauvegarde et restauration

Ils’agit simplement ici de faire un export de sa base de donnée que l’on pourra récupérer derrière pour la réstorer si besoin.

Conclusion   
  
Nous avons vu à travers ce TP la sécursation d’une base de donnée, à travers une bonne gestion des utilisateurs et des droits qui leurs sont alloués. Nous avons utilisé PDO pour empêcher l’injection de code, en utilisant les requêtes préparées. Nous avons également fait des sauvegardes sous forme d’exports de la base de donnée afin de pouvoir la récupérer à l’identique si une mauvaise manipulation venait à arriver.