UNIVERSITÉ Grenoble Alpes

Technologies Web Avancées

Master DDL Parcours DILIPEM

http://lzbk.univ-grenoble-alpes.fr/

Objectifs du cours

- Approfondissement de la compréhension de l'architecture client-serveur
- Notion de
  - Gestion de base de données ;
  - Administration de serveur Web ;
  - Outils de génie logiciel ;
  - Architecture archétypique d'un CMS.

Évaluation

- Travail en groupe (2-4 personnes) 70 % de la note
  - À définir
    - Créer une API
    - Modifier un plugin de CMS
    - Créer un mini CMS
- QCM individuel 30 % de la note

Outils pour le développement

Gestion de versions

Supports / Diaporama

UNIVERSITÉ Grenoble Alpes

Installer moodle

- Se créer une clé ssh et un compte gitlab
  - Supports / Diaporama
- Installer Moodle
  - https://docs.moodle.org/37/en/Git_for_Administrators

UNIVERSITÉ Grenoble Alpes

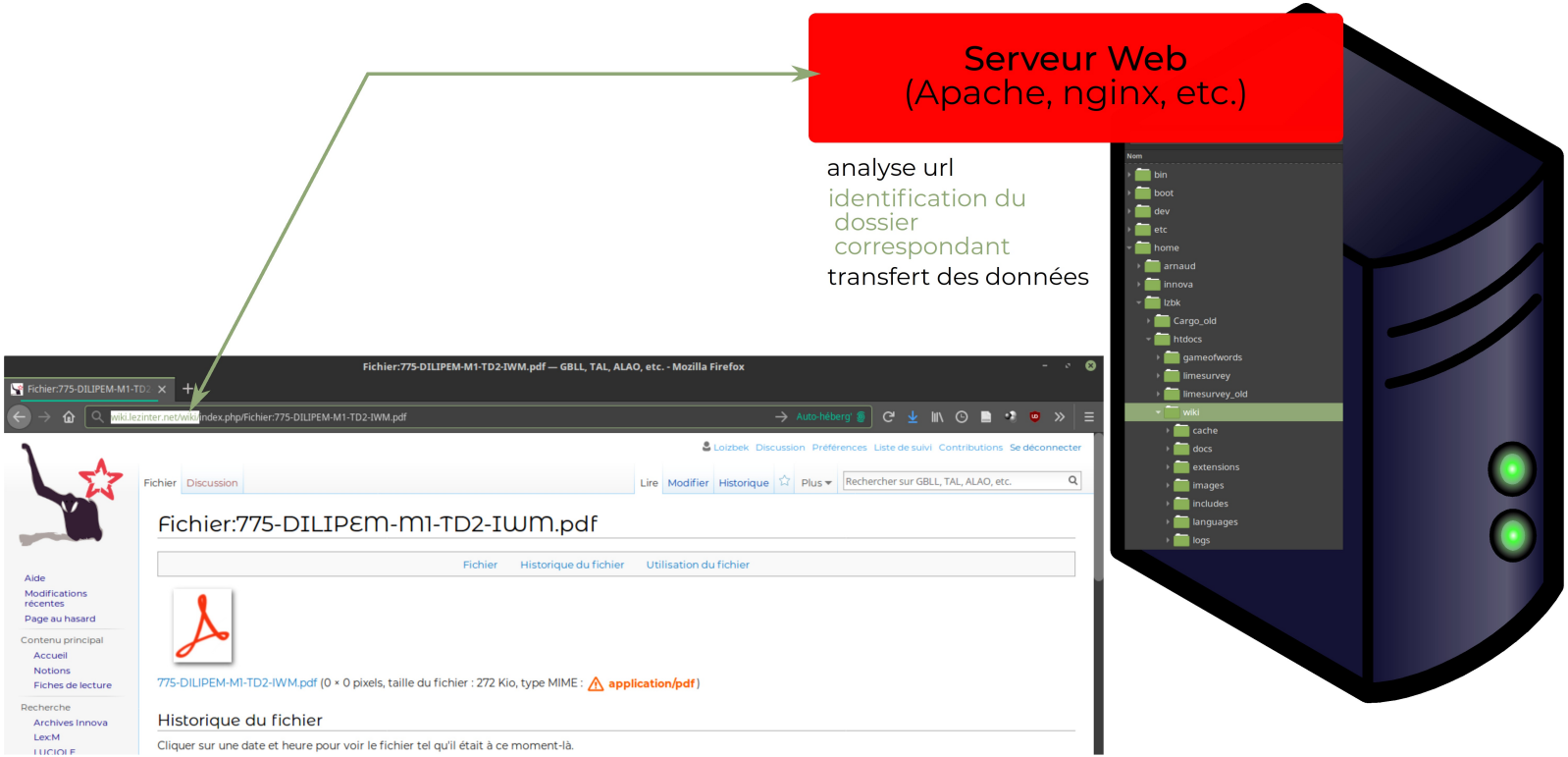
Dossier accessible via le Web

- Dans votre dossier public_html, créez l'arborescence suivante
  - public_html
    - Mon_site
      - Readme.md
      - Images
      - Style
      - Contenus
        - Index.html
    - Secret
      - Photos_persos

- Chargez la page index.html
- Allez directement à la racine de Mon_site
- Naviguez jusqu'à vos photo persos

UNIVERSITÉ Grenoble Alpes

Rappels M1
Serveur Web



Rappels M1
Unified Resource Locator

- https://serveur.uns.it:443/~johndoe/blog/bla/bla.php?a=bla&b=bla

Configuration

```
Alias /~johndoe "/home/johnn/Public_Html/personnel"

<Directory "/home/johnn/Public_Html/personnel">
    Options Indexes FollowSymLinks
    Options FollowSymLinks
    AllowOverride AuthConfig
    Options +Indexes
    Order allow,deny
    Allow from all
    Require all granted
</Directory>
```

- Quelle est le chemin (sur le serveur) du ficher de l'url (Web) sur le serveur Apache configuré comme ci-dessus ?..

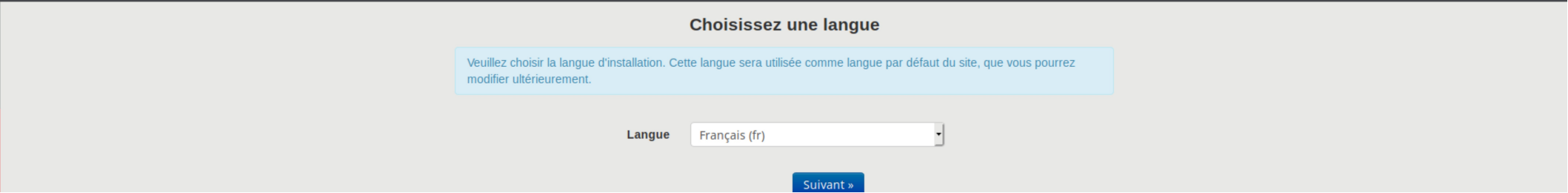
  /home/johnn/Public_Html/personnel/blog/bla/bla.php

UNIVERSITÉ Grenoble Alpes

Étape 1

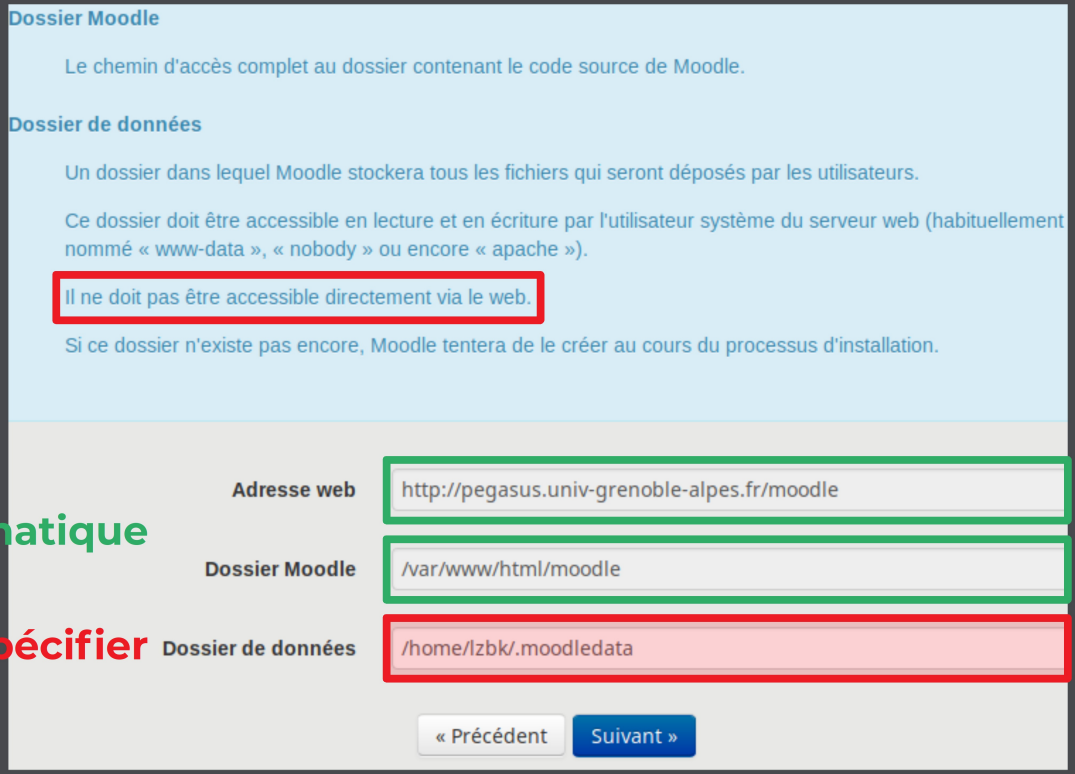
Mettre moodle sur un serveur Web

- Récupérer le code de moodle dans votre dossier « public_html »
  - `ssh user@i3l.univ-grenoble-alpes.fr`
  - Aller dans public_html
  - `git clone https://github.com/moodle/moodle.git`
  - Aller dans le dossier moodle
  - Choisir la bonne branche :
    `git checkout MOODLE_37_STABLE`
- Charger le dossier dans votre navigateur



Étape 2

Configurer les dossiers de données



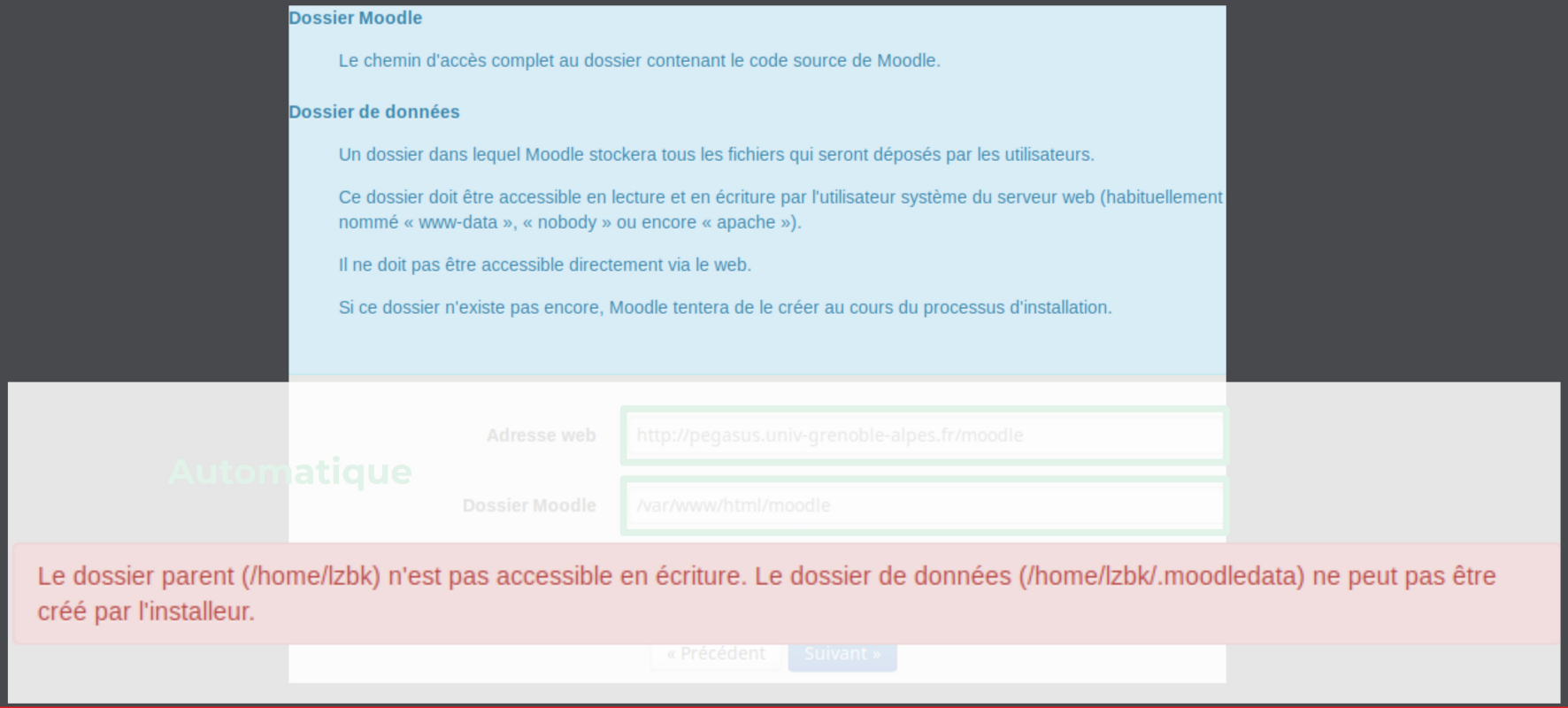
Automatique

À spécifier

UNIVERSITÉ Grenoble Alpes

Étape 2

Configurer les dossiers de données



UNIVERSITÉ Grenoble Alpes

Les droits d'accès dans un système *NIX

```
html % ls -l
total 32
-rw-r--r--  1 root root 10701 juil.  5 15:33 index.html
drwxr-xr-x  6 lzbk www-data  4096 sept.  5 11:25 ki
drwxr-xr-x  8 lzbk www-data  4096 sept.  5 11:04 ki
drwxr-xr-x 16 lzbk www-data  4096 sept. 15 17:51 li
drwxr-xr-x  4 lzbk www-data  4096 oct.   3 09:09 lu
drwxr-xr-x 15 lzbk lzbk      4096 sept. 24 19:41 w
```

- Pour chaque fichier on a :
  - L'utilisateur propriétaire du fichier
  - Le groupe auquel il est associé
  - Les droits d'accès

UNIVERSITÉ Grenoble Alpes

Changer les droits d'un fichier

`ugo=rwx`

- Il y a 3 rôles concernées par les droits :
  - L'utilisateur (**U**ser)
  - Le groupe (**G**roup)
  - Les autres (**O**ther)
- Qui peuvent (ou non)
  - Lire le fichier (**R**ead)
  - Écrire dans le fichier (**W**rite)
  - Exécuter le fichier (e**X**ecute)

```
-rw-r--r--
drwxr-xr-x
```

- Pour changer les droits :

chmod `<droits> nomfichiers`

- Les droits
  - Code octal à trois chiffres
    - 1 chiffre par rôle
  - rôle=droits associés
    - Ex : `ug=rwx, o=r`

Changer le groupe auquel appartient un fichier

- On ne peut affecter un fichier qu'à un groupe auquel on appartient

  `chgrp` `groupe dossier/fichier`

- Le modificateur -R est autorisé.

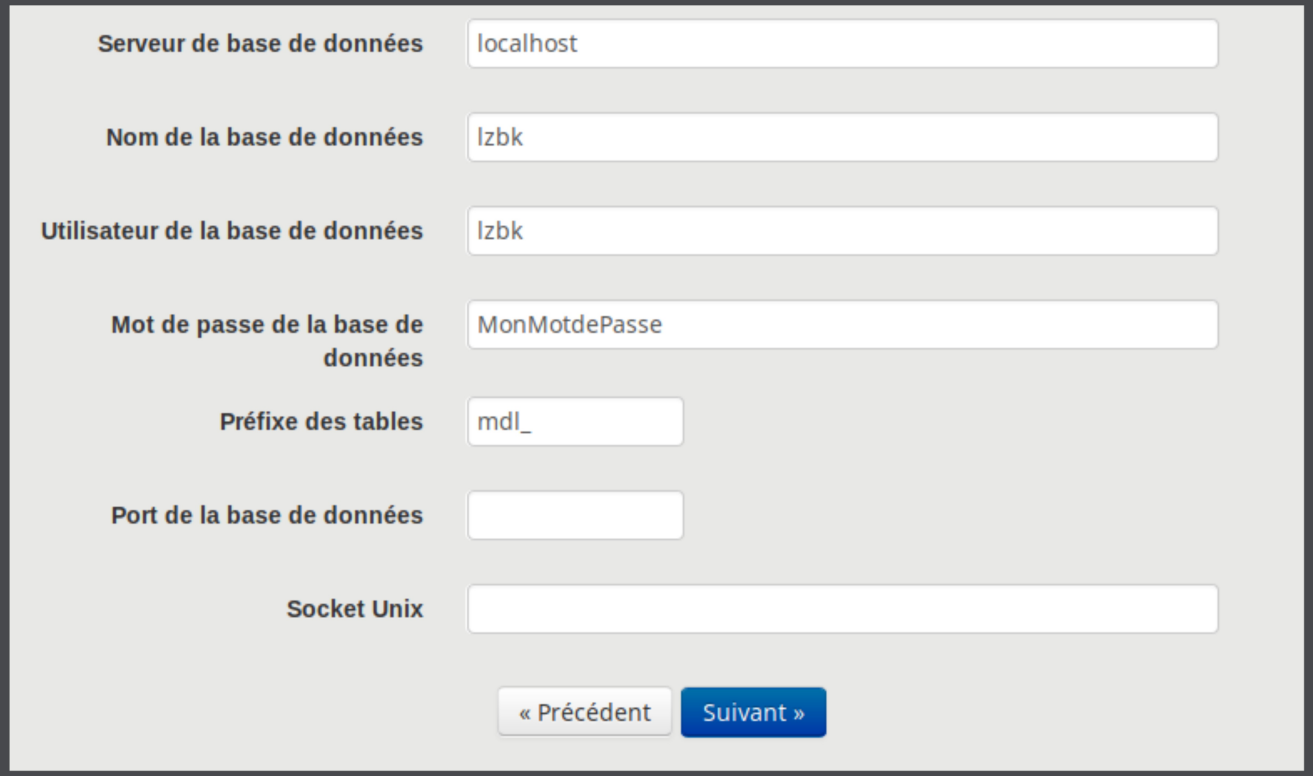
Étape 2

Configurer les dossiers de données

- Créez le dossier caché `.moodledata` dans votre `home`
- Transférez-le au groupe `www-data` (serveur Web)
- Assurez-vous que les membres du groupe ont les droits en écriture sur ce dossier

Étape 3

Configurer la base de données



Étape 3
Configurer la base de données

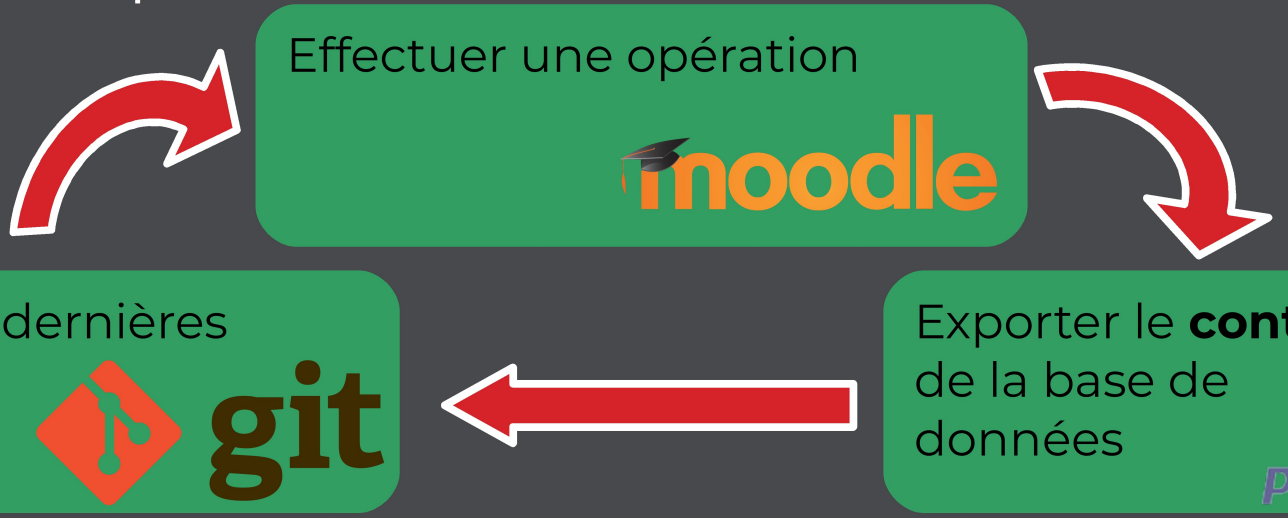
- Connectez-vous à http://i3l.univ-grenoble-alpes.fr/phpmyadmin
- Changez votre mot de passe (anciennement `&votrelogin;`)
- Reportez les informations dans l'interface de configuration de moodle
- Enregistrer le fichier de configuration
  - Cf. gestion des droits
- Toute l'installation doit se faire en une fois… Ça prend un peu de temps, à l'issue de la configuration le système demande
  - Les informations pour la création d'un compte administrateur
  - Les informations globales concernant le moodle (titre, etc.)

Questions...

- Où sont stockés...
  - ...les informations utilisateur ?
  - ...les informations de base d'un cours ?
  - ...les documents ?
  - ...la présence de document dans des cours ?
  - ...les rôles des utilisateurs dans les cours ?

Étape 4
Rétroingénierie

- On veut constater comment les actions effectuées dans moodle affectent les données associées
- Pour cela, un processus itératif :



Étape 4
Rétroingénierie

- Commencer par créer un projet sur http://gricad-gitlab.univ-grenoble-alpes.fr

Observer l'effet des opération moodle sur la base de données

Effectuer une opération

moodle

Exporter le **contenu** de la base de données

phpMyAdmin

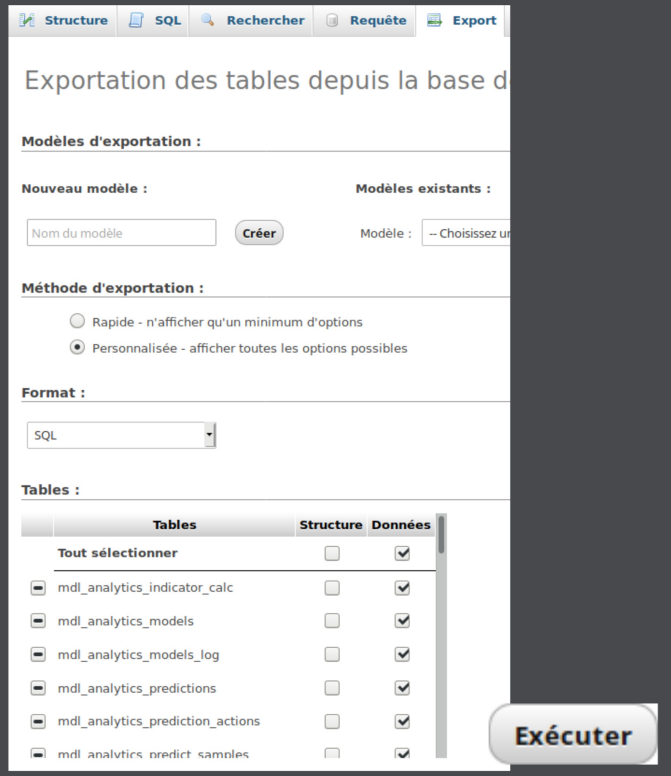
Commiter les dernières modifications

git

Étape 4
Rétroingénierie

Exporter le **contenu** de la base de données

- On exporte toutes les **données** vers un fichier unique qui écrasera la version précédente.
- Par défaut ce fichier s'appellera `nom_de_la_BD.sql`



Étape 4
Rétroingénierie

- Une fois que l'ancienne version du fichier a été remplacée par la nouvelle
- `$ git add nom_de_la_BD.sql`
- `$ git commit -m "opération moodle effectuée"`
- Quand l'utilisateur a fini ses modifications
- `$ git push origin master`

Étape 4
Rétroingénierie

Effectuer une opération

- Créer un commit avec le contenu de la BD après chacune des opérations suivantes

- Créer votre administrateur
- Donner un titre au moodle
- Créer un 2e utilisateur
- Créer un cours
- Créer un 3e utilisateur
- Faire d'un autre utilisateur un enseignant dans ce cours
- Ajouter un document dans le cours
- Ajouter un module au cours
- Ajouter un quizz dans le cours
- Faites répondre au quizz l'un de vos utilisateurs préalablement inscrit au cours en question

UNIVERSITÉ Grenoble Alpes

Étape 4
Rétroingénierie

Observer l'effet des opération moodle sur la base de données

- Chacun doit bien avoir *pushé* tous ses commits sur le serveur
- Gitlab permet de voir les modifications opérées à chaque commit (si elles ne sont pas trop nombreuses)

Assets/GameAssets/Jeux/ScriptableObjects/activity list-string-grades-son.txt

Voir le fichier @ 23d3e953

```
...  ...  @@ -89,4 +89,4 @@
90   90   8,7843,OTHER,1,,,,Beth,QCM,null,Peak
          District#M1.1#M1.2#M1.3#Londres#M4.1#M4.2#Swindon#M4.3#PW#M4.4#Cardiff#A470+#Holyhead#SGC1#SGC#SGC2#Dub
          lin#M7#Limerick#M20+#Skellig#M20+#M7#M1+#Giant#Boat#A82#Horchart#A82#A9M90#Edinburgh,NULL
92        - -1,10142,OTHER,0,,,,Narr.,Narration,null,Londres#Dep#Manche,NULL
     92   + -1,10142,OTHER,0,,,,Narr.,Narration,null,Londres#ParLon#Manche,NULL
```

Quelques clés pour comprendre ce qu'il se passe dans moodle

SGBD et langage de requête

- Un **Système de Gestion de Bases de Données** est un système informatique permettant de stocker, organiser, modifier, rechercher des données
  - MySQL, MariaDB, OpenOffice Base, etc. sont des SGBD relationnelles
- L'interaction avec ces systèmes se fait en général par l'usage d'un langage.
- Pour les BD relationnelles **SQL** est le langage le plus fréquemment utilisé
- *Phpmyadmin* **n'est pas un SGBD**, ni un langage, c'est une interface pour interagir avec le SGBD d'un ordinateur.

Organisation des données

- En général, on va créer une **base de données** différente pour chaque projet
  - Parfois, chaque utilisateur d'un système n'a qu'une base de données (utilisation de préfixes)
- Dans une base de données, les informations sont stockées dans des **tables**
- Les tables spécifient un ensemble de **colonnes**
  (≈liste des champs à remplir)
- Chaque donnée est un **enregistrement** (une ligne) dans une table (spécifiant une valeur — ou non) pour chaque colonne
- L'une des fonctions de SQL est de donner un moyen de spécifier toutes ces structures.

UNIVERSITÉ Grenoble Alpes

Bilan

Les tables de moodle

- Créer votre administrateur
- Donner un titre au moodle
- Créer un 2e utilisateur
- Créer un cours
- Créer un 3e utilisateur
- Faire d'un autre utilisateur un enseignant dans ce cours
- Ajouter un document dans le cours
- Ajouter un module au cours
- Ajouter un quizz dans le cours
- Faites répondre au quizz l'un de vos utilisateurs préalablement inscrit au cours en question

- Regarder en particulier
  - Création utilisateur
  - Création de cours
  - Assignation d'un utilisateur à un cours

Bilan

Les tables de moodle

- Regarder en particulier
  - Création utilisateur
    - `mdl_user`
  - Création de cours
    - `mdl_course`
    - `mdl_enrol`
    - `mdl_course_modules`
    - …
  - Assignation d'un utilisateur à un cours

4 opérations de base

- La manipulation des données (sous-ensemble de SQL) se fonde sur 4 opérations principales :
  - Sélectionner des données `SELECT`
  - Ajouter des données `INSERT`
  - Mettre à jour des données `UPDATE`
  - Supprimer des données `DELETE`

SELECT

- Pour récupérer un ensemble d'informations dans *une* table
  `SELECT` `col1`, `col2` `FROM` `table` `WHERE` condition ;
  - On indique dans quelle table on cherche
  - On sélectionne les colonnes qui nous intéressent (* pour toutes les colonnes)
- Les conditions permettent un ensemble d'opération sur les colonnes ou des valeurs. Par exemple
  - `SELECT * FROM users WHERE users.prenom = 'Jean'`
  - `SELECT nom,prenom FROM users ; WHERE users.age > 40 AND users.age < 51 ;`

Exercice

- Sélectionner les noms, prénoms, noms d'utilisateur et adresse e-mail de tous vos utilisateurs

```
SELECT `firstname`,`lastname`,`username`,`email` FROM `mdl_user`;
```

- Seulement ceux qui ne sont pas l'utilisateur "guest"

```
SELECT `firstname`,`lastname`,`username`,`email`
    FROM `mdl_user`
    WHERE `username`!= 'guest' ;
```

INSERT

- Pour ajouter un ou plusieurs enregistrement dans une table
```
INSERT INTO `table` (`col1`,`col2`) VALUES ('val1','val2'),
('val1','val2') ;
```
  - Les colonnes non indiquées dans la première parenthèse prendront la valeur par défaut ;
  - On peut ajouter plusieurs enregistrements en une requête en séparant les enregistrements entre parenthèses par des virgules
- Exemple
  - ```
    INSERT INTO `users` (`nom`,`prenom`,`age`) VALUES
    ('Jean','Michel',46),('Jean','Micheline',24),('Jean','Killian',1),
    ('Jean','Zinedine',20) ;
    ```
    ```

Exercice

- Ajouter un utilisateur dans votre instance de moodle
  - attention au mot de passe
  - supprimer la colonne `id` de la requête

```
INSERT INTO `mdl_user` (`auth`, `confirmed`, `policyagreed`, `deleted`, `suspended`,
`mnethostid`, `username`, `password`, `idnumber`, `firstname`, `lastname`, `email`,
`emailstop`, `city`, `lang`, `calendartype`, `timezone`, `firstaccess`, `lastaccess`,
`lastlogin`, `currentlogin`, `lastip`, `secret`, `picture`, `url`, `description`,
`descriptionformat`, `mailformat`, `maildigest`, `maildisplay`, `autosubscribe`,
`trackforums`, `timecreated`, `timemodified`, `trustbitmask`)
VALUES ('manual', 1, 0, 0, 0, 1, 'manuel',
'$2y$10$4yOuXECvKj.BjVfhaLOmEu5lzFcYDMmhwP2FbTW1tt/98zdW9dU.y', '', 'Manuel',
'Utilisateur', 'inscriptionManuelle', 0, 'Gregre', 'fr', 'gregorian', '99', 0, 0, 0,
0, '', '', 0, '', '', 1, 1, 0, 2, 1, 0, 1571920146, 1571920146, 0);
```

Exercice

- Connectez-vous avec cet utilisateur :-)
- Exécutez une requête pour trouver l'identifiant de cet utilisateur (sans tous les afficher)

```
SELECT `id`,`username` FROM `mdl_user` WHERE `username`='manuel' ;
```

- Le système a assigné automatiquement un identifiant à l'utilisateur. Allez voir la structure de la table `mdl_user`. Quelle option en est à l'origine ?

Option `AUTO_INCREMENT` pour la colonne `id` (clé/index)

Clés / indexes

- Pour trouver un enregistrement en fonction d'une colonne, le seul moyen est de parcourir toutes les lignes une à une et de regarder la valeur de la colonne…
- Pour gagner du temps, on peut trier la table selon cette colonne…
- Mais que se passe-t-il si on veut pouvoir rechercher dans une table en fonction de plusieurs colonnes selon les cas…
  - Ex : chercher dans un annuaire par nom ou par prénom…
- On peut définir chaque colonne comme un index (ou clé) pour que le système crée ce tri.

Types d'index

- `PRIMARY KEY` (clé primaire)
  - Chaque valeur de la colonne est unique
  - Tous les enregistrements ont une valeur dans cette colonne
  - Une seule clé primaire par table
  - Cette colonne permet d'identifier de manière unique chaque enregistrement
- `UNIQUE KEY` (clé unique)
  - Chaque valeur de la colonne est unique
  - Cette colonne permet d'identifier de manière unique les enregistrements où elle est renseignée
- `INDEX` (index)
  - N'est pas nécessairement unique

UPDATE

- Pour modifier un ou plusieurs enregistrement dans une table

```
UPDATE `table` SET `col1` = 'val1' WHERE `col2` =
'val2' ;
```

  - On modifie tous les enregistrements qui respectent la condition

- Exemple
  -
```
UPDATE `users` SET `age` = `age` + 1
    WHERE `nom` = 'Jean' ;
```

Exercice

- Modifier le nom de famille de l'un de vos utilisateurs moodle

```sql
UPDATE `mdl_user` SET `lastname`='nouveau nom' WHERE `id`= 2;
```

DELETE

- Pour supprimer 1 ou plusieurs enregistrements
  `DELETE FROM `table` WHERE condition`
- On indique quelle table est concernée par la suppression
- On indique sur quel critère on supprime les enregistremens
  - `DELETE FROM `users` WHERE `users`.`nom` = 'Dupont' ;`
  - ```
    DELETE FROM `users`
        WHERE `users`.`age` > 18 AND `users`.`age` < 30 ;
    ```
    ```

Exercice

- Supprimer de moodle l'utilisateur (que vous avez ajouté via l'exercice d'insertion)

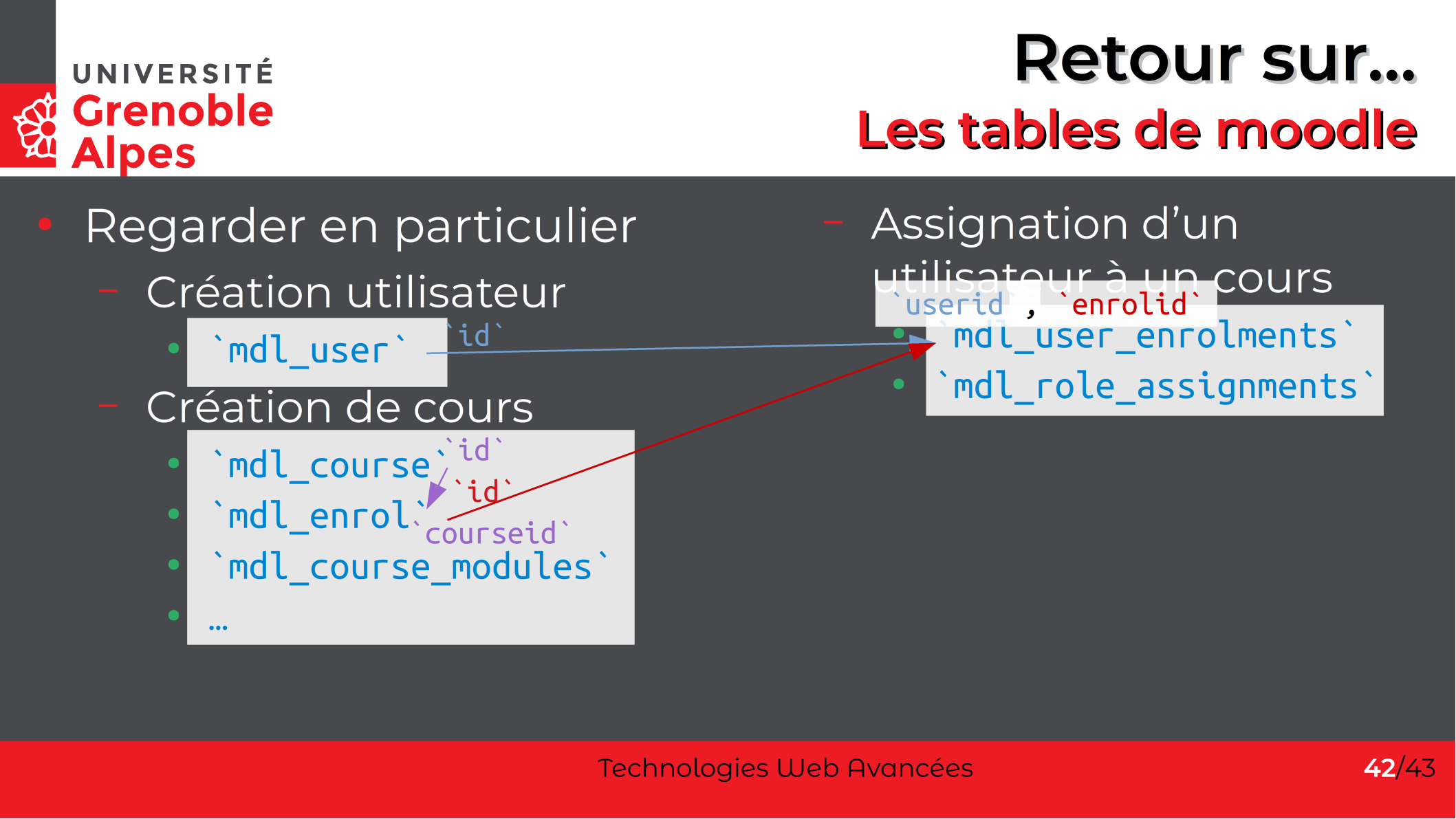
```
DELETE FROM `mdl_user` WHERE `mdl_user`.`id`=4;
```

- Attention : si vous avez fait le test de regarder l'évolution de la base de données lors de la suppression d'un utilisateur, vous comprendrez pourquoi **IL NE FAUT PAS FAIRE CELA DANS UN MOODLE FONCTIONNEL**

UNIVERSITÉ Grenoble Alpes

Retour sur…
Les tables de moodle

- Regarder en particulier
  - Création utilisateur
    - `mdl_user` `id`
  - Création de cours
    - `mdl_course` `id`
    - `mdl_enrol` `courseid`
    - `mdl_course_modules`
    - …
  - Assignation d'un utilisateur à un cours
    `userid` , `enrolid`
    - `mdl_user_enrolments`
    - `mdl_role_assignments`



Jointures et clés étrangères

- `FOREIGN KEY` (clé étrangère)
  - Identifie les valeurs d'une colonne `` `t1`.`c1` `` comme liées à celles d'une colonne `` `t2`.`c2` `` (clé primaire ou unique de la table t2) → rôle logique
  - Permettent de
    - vérifier l'intégriter de la table
    - Déclencher des actions dans `` `t1` `` en cas de certaines opérations dans `` `t2` ``

```
SELECT * FROM `users`
  JOIN `telephones`
    ON `users`.`id` = `telephones`.`userid`
  WHERE `telephones`.`type` = 'Mobile';
```