

SQL avancé

- 1) **Procédures stockées et fonctions**
- 2) Interface entre langage de programmation et SGBD
- 3) Notions d'**administration** de bases de données

Procédures stockées

- Une **procédure stockée** (*stored procedure*) est un ensemble d'instructions SQL précompilées, stockées dans une base de données et exécutées sur demande
- Une procédure stockée peut être lancée par un (compte) utilisateur à condition qu'il en détienne le privilège d'*exécution*

Intérêts

- **simplification** : Code SQL plus simple à comprendre
- **performance** : Économie du temps d'interprétation du code SQL (pré-compilation)
- **sécurité** : Possibilité d'obliger les programmes et les utilisateurs à passer par des procédures stockées sûres (pas d'accès direct aux tables; protection contre les injections SQL...)

Procédures stockées et fonctions

Les fonctions (*user-defined functions*) sont similaires aux procédures stockées.

Différence majeure :

- Une fonction peut être utilisée dans une expression dans une commande SQL (comme les fonctions prédéfinies)
- Une procédure nécessite d'utiliser une commande spécifique

EXECUTE nom_procedure (liste_paramètres_effectifs);

Exemple de définition de procédure

DELIMITER \$\$

CREATE PROCEDURE curseur2()

BEGIN

DECLARE tuple varchar(100) DEFAULT "" ;

DECLARE resultat varchar(2000) DEFAULT "" ;

DECLARE fin BOOLEAN DEFAULT 0;

DECLARE c1 CURSOR FOR SELECT job_title FROM HR.jobs;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = TRUE;

OPEN c1;

REPEAT

FETCH c1 INTO tuple;

set resultat = concat(resultat, tuple);

set resultat = concat(resultat, ' \$\$ ');

UNTIL fin END REPEAT;

SELECT resultat;

CLOSE c1;

END; \$\$

DELIMITER ;

Résultat affiché lors de l'exécution

Public Accountant \$\$\$ Accounting Manager \$\$\$
Administration Assistant \$\$\$ President \$\$\$
Administration Vice President \$\$\$ Accountant \$\$\$
Finance Manager \$\$\$ Human Resources
Representative \$\$\$ Programmer \$\$\$ Marketing
Manager \$\$\$ Marketing Representative \$\$\$ Public
Relations Representative \$\$\$ Purchasing Clerk \$\$\$
Purchasing Manager \$\$\$ Sales Manager \$\$\$ Sales
Representative \$\$\$ Shipping Clerk \$\$\$ Stock Clerk
\$\$\$ Stock Manager \$\$\$ Stock Manager \$\$\$

2) Interface entre langage de programmation et SGBD

- Il est possible à partir d'un programme (PHP, Java, Python) d'ouvrir une connexion avec un serveur de BD, d'envoyer des requêtes et de recevoir les résultats (ou des messages d'erreurs ?)
- API : Interface de programmation d'application
 - Ensemble de fonctions pour réaliser les différentes tâches (ouverture connexion, envoi requête, itération sur les résultats...)
 - API MySQL/PHP : PDO, MySQLi ...

Exemple d'utilisation de l'API MySQLi dans un programme PHP inclus dans un fichier HTML

```
<!DOCTYPE html>
<html>
<head>
<title> Connexion BD mySQL FST </title>
</head>
<body>
<p> <connexion à une BD mySQL </p>
<?php
$hostdb="localhost";
$user = "?";
$password = "?";
$bd="HR";

$connexion = new mysqli($hostdb, $user,
$password, $bd);
```

```
$res3 = $connexion->query (
"INSERT INTO jobs (job_id, job_title) VALUES ('?','?')"
);

if (!$res3)
{echo "Echec lors de l'insertion du tuple : " . $connexion->error ;}
else
{echo "Requête exécutée avec succès";}
?>
</body> </html>
```

3) Introduction à l'administration des Bases de Données

Administrateur (trice) d'une BD : Personne chargée de l'installation et configuration d'un SGBD

- Installer le programme SGBD (attention à la fiabilité de certains sites de téléchargement)
- Créer des bases de données communes et choisir les paramètres (taille, localisation, *charset/interclassement* par défaut...)
- Créer/supprimer des comptes utilisateurs
- Donner/retirer des droits (privilèges) aux utilisateurs

Cas du serveur MySQL dans une composante d'université

- 1) Un(e) ingénieur(e) installe le serveur MySQL sur une machine serveur
- 2) Elle/Il crée un compte pour les enseignant(e)s et leur donne les droits d'administration
- 3) L'enseignant(e) crée un compte pour chaque étudiant(e) avec un mot de passe initial et une BD sur laquelle l'étudiant(e) a tous les privilèges (dont celui de donner des privilèges sur ses éléments...)

Créer un (compte) utilisateur

```
CREATE USER nom_utilisateur  
            IDENTIFIED BY mot_de_passe;
```

Exemples :

```
CREATE USER "titi"@"localhost" IDENTIFIED BY "mdp2" ;
```

```
CREATE USER "toto"@"%" IDENTIFIED BY "mdp1" ;
```


Notion de privilège

Les **privilèges** accordés à un (compte) **utilisateur** déterminent quelles **commandes** l'utilisateur peut effectuer. Ces privilèges peuvent être définis à différents **niveaux** :

- Privilèges administratifs permettent aux administrateurs de détenir des droits globaux pour gérer le serveur de données, applicables à l'ensemble de BDs
- Privilèges au niveau d'une BD : s'appliquent à tous les objets contenus dans une BD ou globalement à toutes les BDs
- Privilèges spécifiques à certains objets d'une BD (table T, vue V, procédure P...) ou à tous les objets d'un certain type (TABLE, PROCEDURE, VIEW) dans une BD ou globalement dans toutes les BDs.

Notion de privilèges

- Privilèges possibles sur une table T ou une vue V existante
 - SELECT : lecture
 - INSERT : ajout de tuples
 - UPDATE : modification de tuples
 - DELETE : suppression de tuples
 - ALL : tous les privilèges
- Plus globalement un privilège porte sur l'utilisation d'une commande à un certain niveau : global/BD/table
- Privilèges possibles sur les procédures stockées
 - CREATE ROUTINE : création d'une procédure/fonction
 - ALTER ROUTINE : modification ou suppression
 - EXECUTE : exécution

Accorder et retirer des privilèges : GRANT et REVOKE

- Accorder des privilèges

GRANT privilège **ON** {*|nomBD.*|nom_table|nom_procedure} **TO** {utilisateur1, utilisateur2...} [**WITH GRANT OPTION**]

Donner à titi le droit de lecture sur la table Commande

```
GRANT SELECT ON maBD.Commande TO "titi"@"localhost"
```

Donner à enzo le droit d'exécuter la procédure proc1 (qui se trouve dans la BD Bdtest)

```
GRANT EXECUTE on PROCEDURE BDtest.proc1 to 'enzo'@'%' ;
```

Donner à enzo le droit de créer des tables (globalement)

```
GRANT CREATE on * to 'enzo'@'%' ;
```

- Retirer des privilèges

REVOKE privilège **ON** {*|nomBD.*|nom_table|nom_procedure} **FROM** {utilisateur1, utilisateur2...}

Retirer le droit de lecture sur Commande à titi

```
REVOKE SELECT ON maBD.Commande FROM "titi"@"localhost"
```

Privilèges possibles dans les commandes GRANT et REVOKE

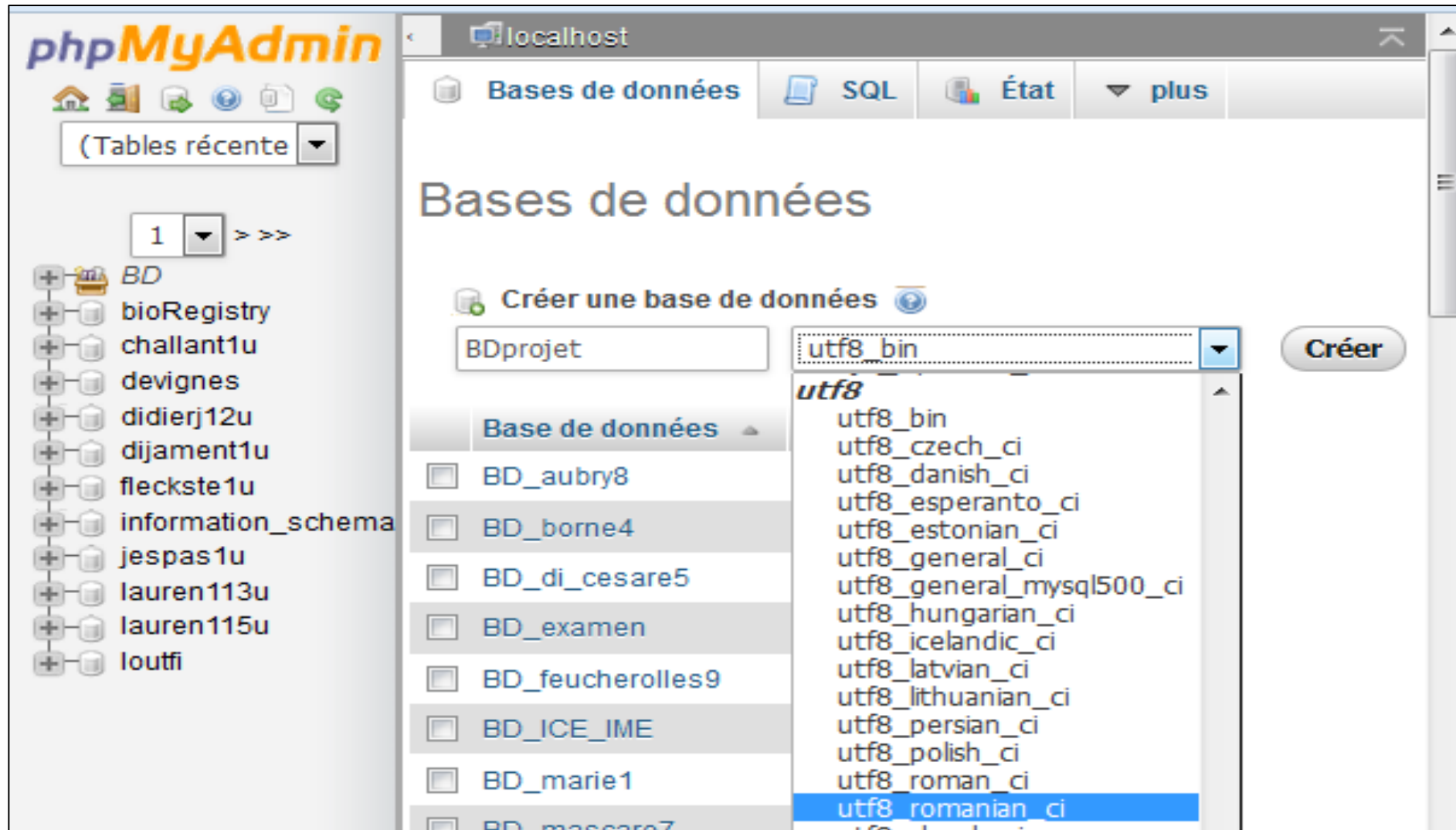
Privilège	Sens et niveaux de privilège
ALL	Grant all privileges at specified access level except GRANT OPTION and PROXY.
ALTER	Enable use of ALTER TABLE. Levels: Global, database, table.
ALTER ROUTINE	Enable stored routines to be altered or dropped. Levels: Global, database, routine.
CREATE	Enable database and table creation. Levels: Global, database, table.
CREATE ROUTINE	Enable stored routine creation. Levels: Global, database.
CREATE USER	Enable use of CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES. Level: Global.
CREATE VIEW	Enable views to be created or altered. Levels: Global, database, table.
DELETE	Enable use of DELETE. Level: Global, database, table.
DROP	Enable databases, tables, and views to be dropped. Levels: Global, database, table.
EXECUTE	Enable the user to execute stored routines. Levels: Global, database, routine.
INSERT	Enable use of INSERT. Levels: Global, database, table, column.
SELECT	Enable use of SELECT. Levels: Global, database, table, column.
SHOW VIEW	Enable use of SHOW CREATE VIEW. Levels: Global, database, table.
SHUTDOWN	Enable use of mysqladmin shutdown. Level: Global.
UPDATE	Enable use of UPDATE. Levels: Global, database, table, column.

Interface d'administration MySQL: Client web phpMyAdmin

phpMyAdmin : interface web développée en PHP et offrant une aide pour l'administration (et l'exploitation) de BDs

- créer de nouvelles bases de données (charset/collation, taille...)
- créer/modifier/supprimer des (comptes) utilisateurs
- gérer les privilèges des utilisateurs
- effectuer des sauvegarde de la structure et/ou des données

Client web phpMyAdmin pour créer une BD



Client web phpMyAdmin pour créer un utilisateur

Bases de données

SQL

État

Utilisateurs

Exporter

Importer

plus

Ajouter un utilisateur

Information pour la connexion

Nom d'utilisateur:

Entrez une valeur:

Client:

Local

localhost

Mot de passe:

Entrez une valeur:

Entrer à nouveau:

Générer un mot de passe:

Générer

Base de données pour cet utilisateur

☐ Créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base

☐ Donner les privilèges passepartout (utilisateur_%)

Privilèges globaux (Tout cocher /Tout décocher)

Commandes utiles en pratique : Import/export de données

Exporter des données de la BD vers un fichier (BD → fichier)

- Avec le client web phpMyAdmin
- A tester : Exporter le contenu d'une de vos tables et noter les différents paramètres d'export personnalisé

Importer des données dans une BD à partir d'un fichier (fichier → BD)

1. Avec le client phpMyAdmin
2. Commande **LOAD DATA** en cas de gros volume de données
 - En mode ligne de commande MySQL

Import de données : commande LOAD DATA

```
LOAD DATA [LOCAL] INFILE      'nom_fichier'  
  [REPLACE|IGNORE] INTO TABLE 'nom_table'  
  [FIELDS  [TERMINATED BY 'caractère']  
           [ESCAPED BY 'caractère' ]]  
  [LINES TERMINATED BY 'caractère']  
  [IGNORE entier LINES]  
  [(noms_attributs) ]
```

