

## R2.06 - Exploitation d'une base de données Cours 6 - Administration MySQL

A. Ridard



## A propos de ce document

- Pour naviguer dans le document, vous pouvez utiliser :
  - le menu (en haut à gauche)
  - l'icône en dessous du logo IUT
  - les différents liens
- Pour signaler une erreur, vous pouvez envoyer un message à l'adresse suivante :  
[anthony.ridard@univ-ubs.fr](mailto:anthony.ridard@univ-ubs.fr)

# Plan du cours

## 1 Utilisateurs

- Classification
- Création, modification, suppression
- Database MySQL VS Schema Oracle

## 2 Privilèges

- Types et niveaux de privilèges
- Principales colonnes de la table mysql.user
- Exemple détaillé

## 3 Compléments

- Rôles
- Accès distants (à venir ou pas)

- 1 Utilisateurs
- 2 Privilèges
- 3 Compléments

## 1 Utilisateurs

- Classification
- Création, modification, suppression
- Database MySQL VS Schema Oracle

## 2 Privilèges

- Types et niveaux de privilèges
- Principales colonnes de la table mysql.user
- Exemple détaillé

## 3 Compléments

- Rôles
- Accès distants (à venir ou pas)

On peut classier les utilisateurs de la manière suivante :

- le **DBA** (DataBase Administrator) qui assure :
  - l'installation et les mises à jour de la base
  - la gestion de l'espace, des utilisateurs et de leurs objets
  - l'optimisation des performances
  - les sauvegardes, restaurations et archivages
  - le contact avec le support technique
- l'**administrateur réseau** qui se charge des accès distants
- les **développeurs** qui conçoivent et implémentent la base
- les **administrateurs des applications** en lien avec la base
- les **utilisateurs** de ces applications



Tous seront des utilisateurs (au sens MySQL) avec des privilèges<sup>a</sup> différents.

---

a. Cette notion sera développée plus loin

## 1 Utilisateurs

- Classification
- **Création, modification, suppression**
- Database MySQL VS Schema Oracle

## 2 Privilèges

- Types et niveaux de privilèges
- Principales colonnes de la table mysql.user
- Exemple détaillé

## 3 Compléments

- Rôles
- Accès distants (à venir ou pas)



L'utilisateur **root** (mot de passe saisi à l'installation) est le DBA que MySQL offre pour effectuer les tâches administratives (créer des utilisateurs par exemple) en ligne de commande (Shell) ou par une console graphique (Workbench)



## Créer un utilisateur<sup>1</sup>

```
CREATE USER ridard@localhost  
IDENTIFIED BY 'iut' — mot de passe de connexion  
;
```

Par défaut, un utilisateur créé n'a aucun droit sur aucune base (à part en lecture écriture sur la base *test* et en lecture seule sur la base *information\_schema*).

---

1. Il faut posséder le privilège **CREATE USER** ou **INSERT** sur la base (système) *mysql* qui stockera l'existence de ce nouvel arrivant.



Pour obtenir des informations sur les utilisateurs, on peut interroger *mysql.user* :



## Lister les utilisateurs

```
SELECT user , host  
FROM mysql . user  
;
```

On obtient alors :

	user	host
▶	mysql.infoschema	localhost
	mysql.session	localhost
	mysql.sys	localhost
	ridard	localhost
	root	localhost

On peut aussi utiliser les fonctionnalités de l'interface graphique :

Local instance MySQL80  
**Users and Privileges**

User Accounts

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
ridard	localhost
root	localhost

**Details for account ridard@localhost**

Login Account Limits Administrative Roles Schema Privileges

Schema	Privileges
bd_ridard	ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREAT

Local instance MySQL80  
**Users and Privileges**

User Accounts

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
ridard	localhost
root	localhost

**Details for account root@localhost**

Login Account Limits Administrative Roles Schema Privileges

Role	Descr	Global Privileges
<input checked="" type="checkbox"/> DBA	grant	<input checked="" type="checkbox"/> ALTER
<input checked="" type="checkbox"/> MaintenanceAdmin	grant	<input checked="" type="checkbox"/> ALTER ROUTINE
<input checked="" type="checkbox"/> ProcessAdmin	rights	<input checked="" type="checkbox"/> CREATE
<input checked="" type="checkbox"/> UserAdmin	grant	<input checked="" type="checkbox"/> CREATE ROUTINE
<input checked="" type="checkbox"/> SecurityAdmin	rights	<input checked="" type="checkbox"/> CREATE TABLESPACE
<input checked="" type="checkbox"/> MonitorAdmin	minim	<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES
<input checked="" type="checkbox"/> DBManager	grant	<input checked="" type="checkbox"/> CREATE USER
<input checked="" type="checkbox"/> DBDesigner	rights	<input checked="" type="checkbox"/> CREATE VIEW
<input checked="" type="checkbox"/> ReplicationAdmin	rights	<input checked="" type="checkbox"/> DELETE
<input checked="" type="checkbox"/> BackupAdmin	minim	<input checked="" type="checkbox"/> DROP
<input checked="" type="checkbox"/> Custom	custo	<input checked="" type="checkbox"/> EVENT



## Renommer un utilisateur <sup>2</sup>

```
RENAME USER ridard@localhost  
        TO aridard@localhost  
;  
  
RENAME USER aridard@localhost  
        TO aridard@194.53.227.12  
;  
  
RENAME USER aridard@194.53.227.12  
        TO ridard@localhost  
;
```

---

2. Il faut avoir le privilège CREATE USER ou UPDATE sur la base *mysql*.



## Modifier le mot de passe d'un utilisateur

```
ALTER USER 'ridard'@'localhost'  
IDENTIFIED BY 'iut_vannes'  
;
```



## Faire expirer le mot de passe d'un utilisateur

```
ALTER USER 'ridard'@'localhost'  
PASSWORD EXPIRE  
;
```



## Supprimer un utilisateur<sup>3</sup>

```
DROP USER ridard@localhost  
;
```

3. Il faut avoir le privilège CREATE USER ou DELETE sur la base *mysql*.

## 1 Utilisateurs

- Classification
- Création, modification, suppression
- Database MySQL VS Schema Oracle

## 2 Privilèges

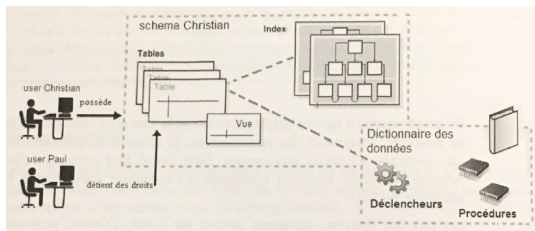
- Types et niveaux de privilèges
- Principales colonnes de la table mysql.user
- Exemple détaillé

## 3 Compléments

- Rôles
- Accès distants (à venir ou pas)



La notion de *database* sur MySQL s'apparente à celle de *schema* sur Oracle c'est à dire à un ensemble d'objets (table, vue, ...) modulo la notion de propriétaire.



## Créer une base <sup>4</sup>

```
CREATE DATABASE bd_ridard  
;
```

4. Il faut avoir le privilège CREATE.



## Sélectionner une base

```
USE bd_ridard — plus besoin de préfixer  
;
```



Pour travailler simultanément avec différentes bases, on préfixe le nom des tables par celui de la base avec la notation pointée : *nomBase.nomTable*



## Modifier le jeu de caractères d'une base<sup>5</sup>

```
ALTER DATABASE bd_ridard  
DEFAULT CHARACTER SET cp850  
;
```



## Supprimer une base<sup>6</sup>

```
DROP DATABASE bd_ridard  
;
```

- 
5. Il faut avoir le privilège ALTER.
  6. Il faut avoir le privilège DROP.



- 1 Utilisateurs
- 2 Privilèges**
- 3 Compléments

## 1 Utilisateurs

- Classification
- Création, modification, suppression
- Database MySQL VS Schema Oracle

## 2 Privilèges

- Types et niveaux de privilèges
- Principales colonnes de la table mysql.user
- Exemple détaillé

## 3 Compléments

- Rôles
- Accès distants (à venir ou pas)

Un **privilège** (sous-entendu *utilisateur*) est un droit :

- d'exécuter une certaine instruction SQL<sup>7</sup> (on parle de privilège **système**)
- de manipuler un certain objet d'une base (on parle de privilège **objet**)



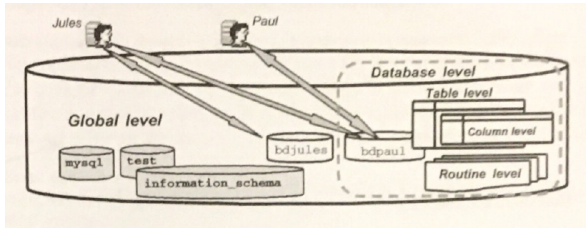
- Les privilèges système diffèrent sensiblement d'un SGBD à un autre<sup>a</sup>
- Les privilèges objet sont les mêmes

---

a. Chez oracle, il y en a plus d'une centaine, MySQL est plus modeste

Voici les différents **niveaux de privilèges**, leur périmètre et leur table de stockage :

- **Global level** : pour toutes les bases du serveur (*mysql.user*)
- **Database level** : pour tous les objets d'une base (*mysql.db* et *mysql.host*)
- **Table level** : pour une table d'une base (*mysql.tables\_priv*)
- **Column level** : pour une colonne d'une table d'une base (*mysql.columns\_priv*)
- **Routine level** : pour les procédures cataloguées<sup>8</sup> (*mysql.procs\_priv*)



## 1 Utilisateurs

- Classification
- Création, modification, suppression
- Database MySQL VS Schema Oracle

## 2 Privilèges

- Types et niveaux de privilèges
- Principales colonnes de la table mysql.user
- Exemple détaillé

## 3 Compléments

- Rôles
- Accès distants (à venir ou pas)

Pour voir les **privilèges** accordés **sur toutes les bases**, on peut interroger `mysql.user` :

### MySQL Privilèges objet (LMD)

```
SELECT user , host , select_priv , insert_priv , update_priv ,  
       delete_priv  
FROM mysql.user  
;
```

### MySQL Privilèges objet (LDD)

```
SELECT user , host , create_priv , drop_priv , index_priv , alter_priv  
FROM mysql.user  
;
```

### MySQL Privilèges système (LCD)

```
SELECT user , host , create_user_priv , grant_priv , show_db_priv  
FROM mysql.user  
;
```



## Privilèges à propos des vues

```
SELECT user , host , create_view_priv , show_view_priv  
FROM mysql.user  
;
```



## Privilèges à propos des procédures cataloguées

```
SELECT user , host , create_routine_priv , alter_routine_priv ,  
        execute_priv  
FROM mysql.user  
;
```



## Privilèges à propos des restrictions d'utilisateur

```
SELECT user , host , max_questions , max_updates , max_connections ,  
        max_user_connections  
FROM mysql.user  
;
```

## 1 Utilisateurs

- Classification
- Création, modification, suppression
- Database MySQL VS Schema Oracle

## 2 Privilèges

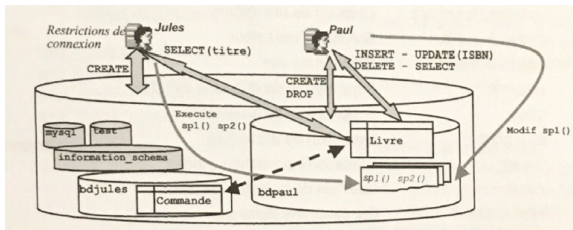
- Types et niveaux de privilèges
- Principales colonnes de la table mysql.user
- Exemple détaillé

## 3 Compléments

- Rôles
- Accès distants (à venir ou pas)



Voici la situation que nous allons développer :



- L'instruction **GRANT** permet d'attribuer un (ou plusieurs) privilège(s) à un (ou plusieurs) bénéficiaire(s)
- L'utilisateur qui exécute cette commande doit avoir reçu lui-même le droit de transmettre ces privilèges (avec la directive **GRANT OPTION**)
- Dans le cas de *root* que l'on va prendre pour notre exemple, aucun problème car il a tous les droits



## Préparation de l'exemple

```
CREATE USER Paul@localhost IDENTIFIED BY 'info' ;  
CREATE USER Jules@localhost IDENTIFIED BY 'stid' ;  
CREATE DATABASE bdpaul ;  
CREATE DATABASE bdjules ;
```



### Privilège système *database level*

```
GRANT CREATE, DROP  
ON bdpaul.*  
TO 'Paul'@'localhost'  
;
```



### Privilège objet *table level*

```
GRANT INSERT, SELECT, DELETE, UPDATE(ISBN)  
ON bdpaul.Livre  
TO 'Paul'@'localhost'  
;
```



### Privilège système *table level*

```
GRANT ALTER  
ON bdpaul.Livre  
TO 'Paul'@'localhost'  
;
```

 Privilège objet *column level*

```
GRANT SELECT(titre)
ON bdpaul.Livre
TO 'Jules'@'localhost'
WITH GRANT OPTION — Jules pourra retransmettre ce droit
;
```

 Privilège système *global level*

```
GRANT CREATE ON *.*
TO 'Jules'@'localhost'
;
```



## Privilège système *database level*

```
GRANT USAGE ON bdpaul.*  
  TO 'Jules'@'localhost'  
  WITH  
    MAX_QUERIES_PER_HOUR 50  
    MAX_UPDATES_PER_HOUR 20  
    MAX_CONNECTIONS_PER_HOUR 6  
    MAX_USER_CONNECTIONS 3  
;
```



## Privilège objet *table level*

```
GRANT REFERENCES ON bdpaul.Livre  
  TO 'Jules'@'localhost'  
;
```

Pour voir les privilèges, on peut exécuter :



```
SHOW GRANTS FOR 'Jules'@'localhost' ;
```

On peut aussi interroger *mysql.db*, *mysql.tables\_priv* et *mysql.columns\_priv*.



### Table *mysql.db*

```
SELECT CONCAT(user, '@', host) compte, create_priv, drop_priv,  
       alter_priv  
FROM mysql.db ;
```



### Table *mysql.tables\_priv*

```
SELECT CONCAT(user, '@', host) compte, CONCAT(db, '.', table_name)
       objet, grantor, table_priv
FROM mysql.tables_priv
;
```



### Table *mysql.columns\_priv*

```
SELECT CONCAT(user, '@', host) compte, CONCAT(db, '.', table_name)
       objet, column_name, column_priv
FROM mysql.columns_priv
;
```



- La table `mysql.procs_priv` sera utile au S3
- La table `mysql.host` sera étudiée dans la section "Accès distants"





## Révoquer un privilège<sup>9</sup>

```
REVOKE CREATE
  ON bdpaul.*
  FROM 'Paul'@'localhost'
;

REVOKE INSERT, UPDATE(ISBN)
  ON bdpaul.Livre
  FROM 'Paul'@'localhost'
;

GRANT USAGE
  ON bdpaul.*
  TO 'Jules'@'localhost'
  WITH
    MAX_QUERIES_PER_HOUR 50
    MAX_UPDATES_PER_HOUR 20
;

REVOKE ALL PRIVILEGES, GRANT OPTION FROM 'Jules'@'localhost'
;
```

9. Il faut le détenir avec l'option WITH GRANT OPTION.

- 1 Utilisateurs
- 2 Privilèges
- 3 Compléments**

## 1 Utilisateurs

- Classification
- Création, modification, suppression
- Database MySQL VS Schema Oracle

## 2 Privilèges

- Types et niveaux de privilèges
- Principales colonnes de la table mysql.user
- Exemple détaillé

## 3 Compléments

- Rôles
- Accès distants (à venir ou pas)

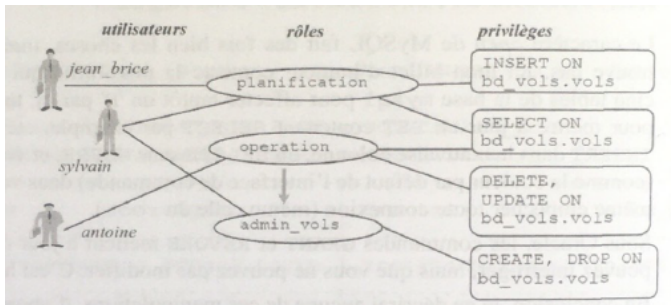


- Un rôle est un ensemble nommé de privilèges (système ou objet)
- Il est accordé à un (ou plusieurs) utilisateur(s) ou autre(s) rôle(s)
- Ce mécanisme facilite la gestion des privilèges

Voici la chronologie des actions à entreprendre pour travailler avec des rôles :

- créer le rôle
- alimenter le rôle avec des privilèges (système ou objet)
- attribuer le rôle à d'autres rôles ou à des utilisateurs
- faire évoluer le rôle dans le temps en ajoutant ou en enlevant des privilèges

Voici la situation que nous allons développer :



MySQL

## Créer et alimenter des rôles

```
CREATE ROLE 'planification', 'operation', 'admin_vols'  
;  
  
GRANT INSERT, SELECT  
  ON bd_vols.vols  
  TO 'planification'  
;  
  
GRANT SELECT, DELETE, UPDATE  
  ON bd_vols.vols  
  TO 'operation'  
;  
  
GRANT CREATE, DROP  
  ON bd_vols.*  
  TO 'admin_vols'  
;  
  
GRANT 'admin_vols'  
  TO 'operation'  
;
```



## Affecter des rôles à des utilisateurs

```
GRANT 'planification'  
  TO 'jean_brice'@'localhost', 'sylvain'@'localhost'  
;  
  
GRANT 'operation'  
  TO 'antoine'@'localhost'  
;  
  
GRANT 'admin_vols'  
  TO 'sylvain'@'localhost'  
;
```



Affecter un rôle à un utilisateur ne rend pas ce rôle effectif pour toutes les connexions, mais seulement l'actuelle.

MySQL

## Valider des rôles

```
SET DEFAULT ROLE ALL — NONE annulerait les précédentes
    assignations
    TO 'jean_brice'@'localhost', 'sylvain'@'localhost', 'antoine'@
    'localhost'
;

SELECT CURRENT_ROLE()
;
```

MySQL

## Modifier et supprimer des rôles

```
GRANT INDEX
    ON bd_vols.*
    TO 'admin_vols'
;

REVOKE DELETE
    ON bd_vols.vols
    FROM 'operation'
;

DROP ROLE 'planification', 'operation', 'admin_vols'
;
```



## 1 Utilisateurs

- Classification
- Création, modification, suppression
- Database MySQL VS Schema Oracle

## 2 Privilèges

- Types et niveaux de privilèges
- Principales colonnes de la table mysql.user
- Exemple détaillé

## 3 Compléments

- Rôles
- Accès distants (à venir ou pas)