MySQL : Sécurité

Introduction

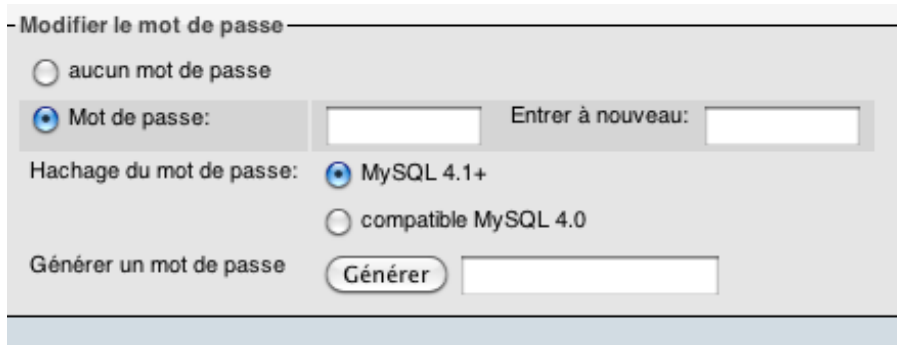
Pourquoi sécuriser ?

- Imaginez une altération des N° de CB
- Sabotage ? Maladresse ?
- C'est un travail collectif (pas que le RSSI)
  Responsable de la Sécurité des Système d'information
- L'installation par défaut est permissive
  - Besoin d'agir -> Sécuriser l'installation

Sécuriser le serveur

Sécuriser l'installation

- Contrôler les droits (voir plus loin)
- Mettre un mot de passe à root
- Créer plusieurs utilisateurs



Sécuriser l'installation

	Utilisateur	Serveur	Mot de passe	Privilèges globaux 1	«Grant»	Action
☐	N'importe quel	%	--	USAGE	Non	
☐	N'importe quel	localhost	Non	USAGE	Non	
☐	root	localhost	Oui	ALL PRIVILEGES	Oui	

- Supprimer les comptes anonymes
  -> Pas de mot de passe
- Supprimer le schéma test
  -> Risque saturation de l'espace de stockage

Mise en pratique

Modification de mot de passe

Query 1

```
1  set password for root@localhost=password('Azerty11');
```

Dans la table user de la base mysql le mot de passe est crypté

```
1  select user,host,password
2  from mysql.user
3  where user='root';
```

100% 26:1

Overview | Output | Snippets | Query 1 Result

Filter: Fetched 1 records.

user	host	password
root	localhost	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B

Mise en pratique

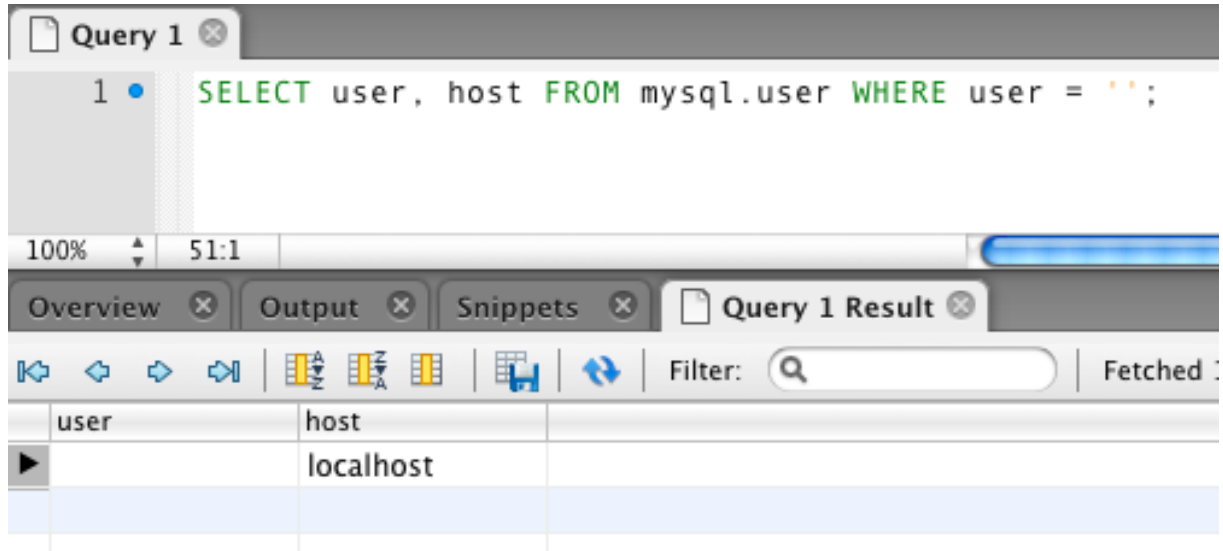
On peut renommer le compte root afin de le rendre plus difficile à trouver

Query 1

```
1 • RENAME USER 'root'@'localhost' TO 'super'@'localhost';
```

Mise en pratique

Voir les comptes anonymes



La requête nous dit qu'il y a un compte anonyme sur le host local

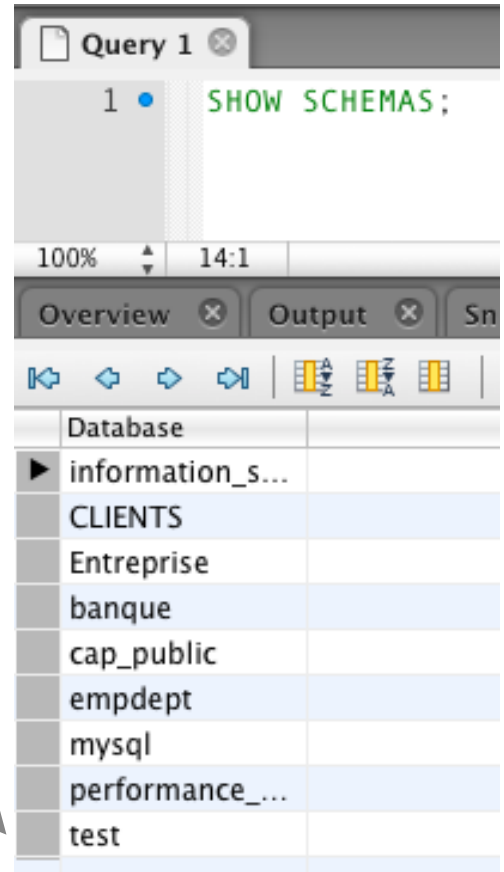
Supprimer les comptes anonymes

```
DROP USER ''@localhost;
```

Mise en pratique

Voir si il y a la base test

SHOW SCHEMAS;



Si on veut supprimer une base :

DROP SCHEMA test;

Gestion des utilisateurs

Gestion des utilisateurs

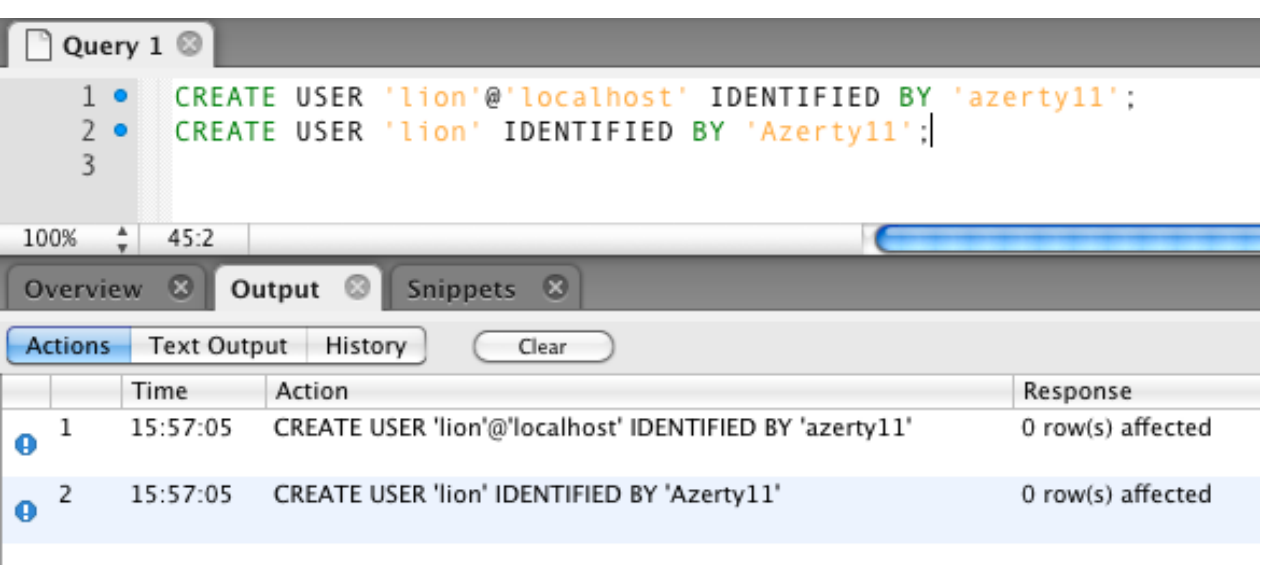
- Se fait dans la table user de la base mysql
- Un utilisateur est décrit via :
  - user@host
  - on peut donc avoir plusieurs root
    Ex : root@localhost, root@192.178.23.1
    Il s'agit ici de deux utilisateurs différents

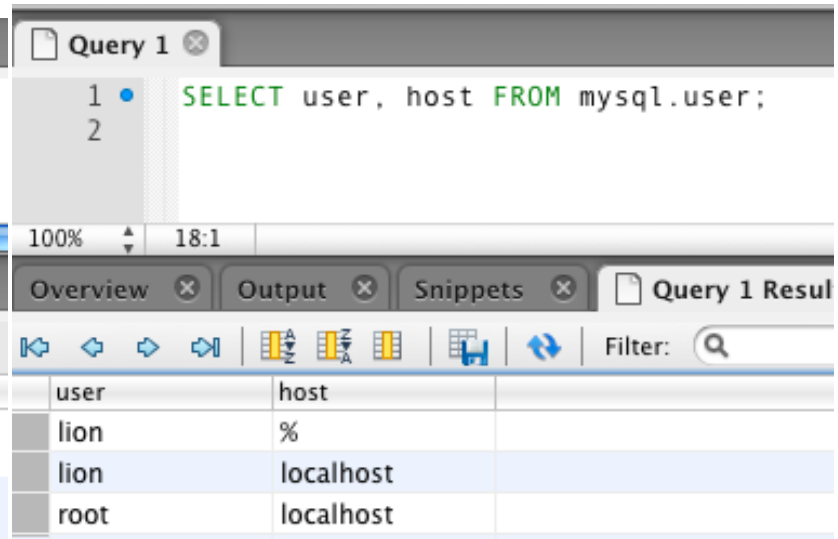
Gestion des utilisateurs

- La création d'un compte se fait avec
  CREATE USER
  Exemple :

- La suppression d'un utilisateur avec
  DROP USER
  Exemple : DROP USER 'olivier'@'localhost';

Gestion des utilisateurs





On a deux comptes : un compte si on utilise un client local dans tous les autres cas ça sera l'autre @'%'

Gestion des utilisateurs

Notion de joker -> %

On peut utiliser le joker afin d'affiner le filtrage. Par exemple :

CREATE USER 'Paul'@'%.bts.sio.net'

On peut se connecter en tant que Paul sur les machines dont le nom se termine par .bts.sio.net
Autre exemple avec l'adresse IP :

CREATE USER 'Paul'@'192.168.%'

Donner des droits

Commande GRANT

- Elle permet de donner des droits
- Créer le compte si il n'existe pas
- Syntaxe :

```
GRANT type_de_droits ON portée_des_droits
TO compte_utilisateur;
```

- Type de droits : SELECT, PROCESS (voir les threads)...
- Portée_des_droits : tout le serveur, une base de données, une table ?

Commande GRANT

- On distingue 4 types de droits :
  - Les droits d'administration

    ➡GRANT OPTION : permet de transmettre ces droits

    ➡SHUTDOWN : Arrêter le serveur

    ➡REPLICATION SLAVE : Récupérer les données du journal maitre

    ➡...
  - Les droits sur les schémas (BD)

    ➡ALTER, CREATE, DELETE, DROP, INSERT, SELECT, UPDATE...
  - Les droits au niveau des tables

    ➡ALTER, DROP, TRIGGER, INDEX, ...
  - Les droits au niveau des routines

    ➡EXECUTE, CREATE ROUTINE, ALTER ROUTINE

Petits tests

GRANT : affecter des privilèges

- ```
  GRANT ALL PRIVILEGES
  ON *.*
  TO admin ;
  ```

- Le script suivant donne tous les droits sur l'ensemble du serveur :

- ```
  GRANT REPLICATION SLAVE
  ON *.*
  TO replicateur@127.0.0.1 ;
  ```

- Le script suivant donne les droits nécessaires au compte replicateur (local) pour la réplication

GRANT : affecter des privilèges

- ```
  GRANT ALL
  ON bddgestion.*
  TO admin ;
  ```

- Le script suivant donne tous les droits (sauf le droit d'accorder des droits) sur toutes les tables de la base *bddgestion* :

- ```
  GRANT SELECT (nom, prénom),
  UPDATE (nom, prénom)
  ON bddgestion.user
  TO visiteur ;
  ```

- Le script suivant donne les droits (à visiteur) SELECT et UPDATE uniquement sur les colonnes *nom* et *prénom* de la table *user*.

GRANT : affecter des privilèges

- ```
  GRANT SELECT
  ON bddgestion.produit
  TO visiteur ;
  ```

- ```
  GRANT SELECT, UPDATE,
  DELETE, INSERT
  ON bddgestion.produit,
  bddgestion.gamme
  TO admin, visiteur;
  ```

- ```
  GRANT SELECT
  ON bddgestion.*
  TO visiteur ;
  ```

Le script suivant donne le droit à **visiteur** d'exécuter la commande **SELECT** sur la table **produit** de la base **bddgestion** :

Le script suivant montre qu'on peut donner en une seule commande GRANT plusieurs privilèges, sur plusieurs tables, à plusieurs utilisateurs :

nom-base.* permet de désigner toutes les tables d'une base de données :
  ```

GRANT : affecter des privilèges

- ```
  GRANT CREATE ROUTINE,
  EXECUTE ON ps.*
  TO 'create_ps'@'localhost';
  ```

  Donne le droit de créer et d'exécuter des procédures stockées dans le schéma ps

- ```
  GRANT SHUTDOWN
  ON *.*
  TO 'arret_serveur'@'localhost';
  ```

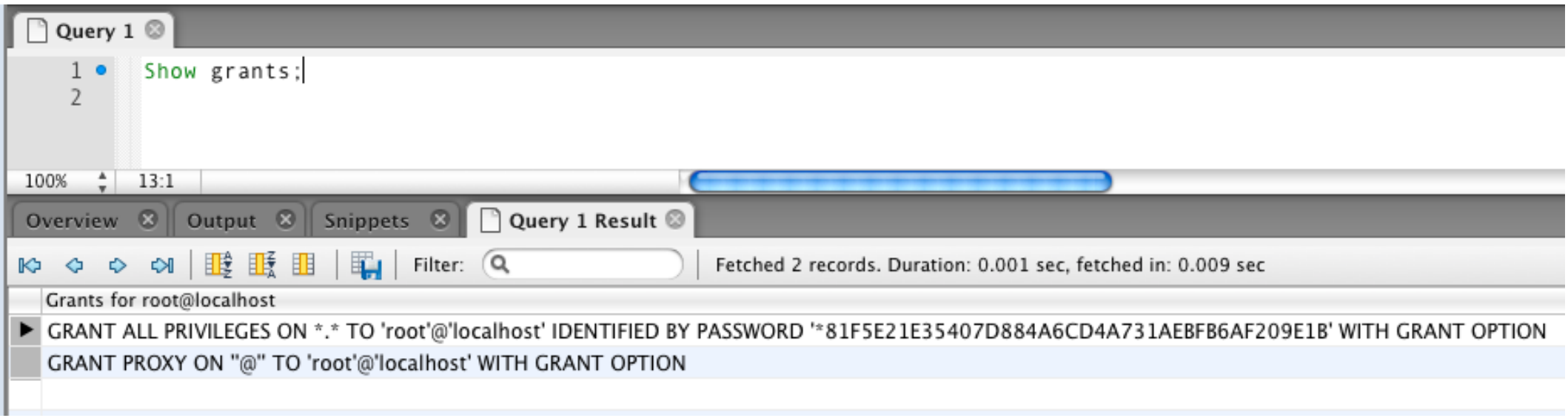
  Donne le droit d'arrêter le serveur MySQL à arret_serveur (local)

Visualiser ses droits

Visualiser ses droits

SHOW GRANTS;

- Permet de visualiser ses propres droits
- Ceux des autres si on a les droits



Supprimer des droits

REVOKE:Retirer des droits

Syntaxe:

- REVOKE type_de_droits ON portée_des_droits FROM compte;

Remarque :

Revoke retire les droits mais ne supprime pas l'utilisateur (-> DROP USER)

REVOKE : retirer des privilèges

- ```
  REVOKE UPDATE
  ON bddgestion.user
  FROM visiteur ;
  ```

  Le script suivant retire le droit de mise à jour sur la table user :

- ```
  REVOKE ALL PRIVILEGES
  ON bddgestion.user
  FROM visiteur ;
  ```

  Le script suivant retire tous les privilèges de la table *user* à visiteur.

- ```
  REVOKE GRANT OPTION
  FROM secretaire ;
  ```

  Le script suivant conserve les privilèges, mais retire le droit de les accorder à d'autres utilisateurs :
  ```

TP

- Création d'un compte Admin sur photoForYou : **adminPhoto** nécessaire à la gestion de la base (il a tous les droits nécessaires mais que sur la base PhotoForYou). Ce compte se connecte en local.
- Création d'un compte pour les pages Web pour le code PHP (Quels droits sont nécessaires sur la base PhotoForYou ?). **webPhotoForYou**
- Remarques : une bonne pratique est de retirer tous les droits et d'accorder après que ceux nécessaires.