M106 — Rôles et autorisations

F. Mauron

EPAI

05 décembre 2023

Plan

Plan

Définition du RBAC

Le RBAC (Role-Based Access Control) est un système de contrôle d'accès où les droits et permissions ne sont pas attribués directement aux utilisateurs individuels, mais plutôt à des rôles. Les utilisateurs sont ensuite assignés à ces rôles, héritant des permissions associées à ces rôles.

Principes fondamentaux du RBAC

- **Rôles :**
  - **Définition :** Un rôle représente un ensemble de permissions liées à des tâches spécifiques.
  - **Exemples :** Administrateur, éditeur, utilisateur standard, auditeur.
- **Permissions :**
  - **Association avec les rôles :** Les permissions spécifient ce que l'on peut faire (par exemple, lire, écrire, supprimer).
  - **Granularité :** Les permissions peuvent être très granulaires, permettant un contrôle précis de l'accès.

Avantages du RBAC

- **Simplification de la gestion des autorisations :** La gestion des permissions devient plus simple car elles sont gérées au niveau des rôles plutôt qu'individuellement pour chaque utilisateur.
- **Réduction des risques de sécurité :** En limitant l'accès aux fonctionnalités nécessaires pour un rôle donné, le RBAC réduit le risque d'abus ou d'erreur.
- **Conformité et audit facilités :** Le RBAC aide à respecter les réglementations de conformité et simplifie les audits de sécurité.

Mise en oeuvre du RBAC

- **Analyse des besoins d'accès :** Identifier les différentes tâches au sein de l'organisation et les regrouper en rôles.
- **Définition des rôles et permissions :** Créer des rôles basés sur les besoins d'accessibilité et associer les permissions appropriées.
- **Attribution des rôles aux utilisateurs :** Assigner les utilisateurs aux rôles correspondants à leurs responsabilités.
- **Révision et mise à jour :** Réviser périodiquement les rôles et les permissions pour s'assurer qu'ils restent pertinents et sécurisés.

Bonnes pratiques pour le RBAC

- **Principe de moindre privilège :** Attribuer uniquement les permissions nécessaires pour accomplir les tâches.
- **Surveillance et audit :** Suivre l'utilisation des permissions et réaliser des audits réguliers pour détecter toute utilisation inappropriée.
- **Flexibilité et évolutivité :** Concevoir le système RBAC pour qu'il soit flexible et évolutif afin de s'adapter aux changements organisationnels.

Plan

Commandes du Langage de Contrôle de Données (DCL) pour la gestion des utilisateurs et des rôles

Utiliser le DCL pour gérer la sécurité des bases de données en contrôlant les accès des utilisateurs et la définition des rôles.

Plan

Gestion des utilisateurs

Créer un utilisateur :

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

Supprimer un utilisateur :

```
DROP USER [nom_utilisateur];
```

Gestion des rôles

Créer un rôle :

```
CREATE ROLE [nom_role];
```

Supprimer un rôle :

```
DROP ROLE [nom_role];
```

Attribution de rôles et privileges

Attribuer un rôle à un utilisateur :

```
GRANT [nom_role] TO [nom_utilisateur];
```

Révoquer un rôle d'un utilisateur :

```
REVOKE [nom_role] FROM [nom_utilisateur];
```

Attribuer des privileges :

```
GRANT [privilege] ON [objet] TO [nom_utilisateur/role];
```

Révoquer des privileges :

```
REVOKE [privilege] ON [objet] FROM [nom_utilisateur/role];
```

Vous trouverez la liste des privilèges **ici**

Plan

Bonnes pratiques

- **Sécurité des mots de passe :** Assurer que les mots de passe des utilisateurs sont forts et sécurisés.
- **Principe de moindre privilège :** Attribuer uniquement les privilèges nécessaires pour les tâches spécifiques.
- **Audit et surveillance :** Surveiller régulièrement l'utilisation des privilèges pour détecter toute activité suspecte.

Plan

Exemple I/IV

```sql
CREATE DATABASE IF NOT EXISTS test;
CREATE ROLE read_only;
GRANT SELECT ON test.* TO read_only;
GRANT USAGE ON test.* TO read_only;
-- SHOW GRANTS FOR read_only;

USE test;

CREATE TABLE Articles (
    articleID INT AUTO_INCREMENT PRIMARY KEY,
    articlename VARCHAR(255) NOT NULL,
    price DECIMAL(6,2)
);
```

Exemple II/IV

```
INSERT INTO Articles (articlename, price) VALUES
('RAM 8GB', 128.40),
('CPU Intel I9', 899.00),
('Power 850W', 232.80);

CREATE USER test_user@'%' IDENTIFIED BY '1234';
-- SHOW GRANTS FOR test_user@'%';
GRANT read_only TO test_user@'%';
-- SHOW GRANTS FOR test_user@'%';

SET DEFAULT ROLE read_only FOR test_user@'%';
FLUSH PRIVILEGES;
```

Exemple III/IV

Ouvrez un terminal powershell ou bashshell

```
$ mysql -h svr-m164-XY-mysql.lab.epai-ict.ch -u test_user -p
$
mysql> use test;
ERROR 1044 (42000): Access denied for user 'test_user'@'%' to database 'test'

USE test;
```

Exemple IV/IV

- Essayer de lister les tables
- Essayer de lister le contenu de la table Articles
- Essayer de créer une nouvelle table, de modifier et desupprimer la table Article
- Essayer les commandes suivantes et déterminez leur fonction :
  - SELECT current_role();
  - SELECT current_user();
  - SHOW GRANTS;

Vous pouvez télécharger le script **ici**