Адміністрування та безпека у MySQL

1. Облікові записи користувачів

1.1 Реєстрація користувача

Обліковий запис користувача створюється командою:

CREATE USER <Ідентифікатор користувача>

[IDENTIFIED BY [PASSWORD] '<Пароль >'];

Обов'язковим параметром є ідентифікатор нового користувача. Якщо не заданий параметр ***IDENTIFIED BY***, то використовується порожній пароль.

Параметр ***PASSWORD*** слід вказати у випадку, коли вводиться не реальний, а зашифрований пароль. Отримати зашифроване значення з реального пароля можна за допомогою функції

PASSWORD ('<Реальний пароль>')

Наприклад, команда

CREATE USER ‘anna' IDENTIFIED BY 'annapassword';

створює обліковий запис для користувача з ім'ям ***anna***, що підключається з будь-якого комп'ютера, і встановлює для облікового запису пароль ***annapassword***. Команда

CREATE USER 'anna’@’localhost' IDENTIFIED BY

PASSWORD ‘*3C7F72EAE78BC95AAFBFD21F8741C24A0056C04B';

створює обліковий запис для користувача ***anna***, що підключається з локального комп'ютера, і встановлює як пароль значення ***annalocpassword*** (оскільки функція ***PASSWORD ('annalocpassword')*** повертає значення

3C7F72EAE78BC95AAFBFD21F8741C24A0056C04B).

1.2 Установка пароля

Для установки пароля призначена команда

SET PASSWORD [FOR <Ідентифікатор користувача>] = PASSWORD ('<Пароль>');

Параметрами команди є ідентифікатор облікового запису користувача і новий пароль для запису. Замість функції PASSWORD(), що шифрує реальний пароль, можна ввести зашифрований пароль. Команди

SET PASSWORD FOR 'anna‘@'%' = PASSWORD (‘newannapassword'); і

SET PASSWORD FOR 'anna’@'%' = ‘*006B99DE1BDA1BE6E1FFF714E764A8FAB0E614DF';

установлюють пароль ***newannapassword*** для користувача ***anna***, що підключаться з будь-якого комп'ютера.

1.3 Видалення користувача

Видалити обліковий запис можна за допомогою команди

DROP USER <Ідентифікатор користувача>;

Після видалення користувач не може підключатися до сервера MySQL. Однак якщо на момент видалення користувач був підключений до сервера, то з'єднання не переривається.

Разом з обліковим записом видаляються всі привілеї доступу для цього запису.

1.4 Перегляд облікових записів

Для отримання інформації про зареєстро-ваних користувачів виконуємо запит до таблиці ***user*** (Користувач) системної бази даних ***mysql***, наприклад

```
SELECT * FROM mysql.user;
```

Перші три стовпці таблиці ***user*** - ***Host***, ***User*** і ***Password***.

2. Система привілеїв доступу

2.1 Загальні відомості про привілеї доступу

В MySQL використовуються такі типи привілеїв:

- ***ALL[PRIVILEGES]*** - надає всі привілеї, крім GRANT OPTION, для вказаної області дії;
- ***ALTER*** - дозволяє виконання команд ALTER DATABASE і ALTER TABLE;
- ***CREATE*** - виконання команд CREATE DATABASE і CREATE TABLE;
- ***CREATE USER*** - виконання команд CREATE USER, DROP USER, RENAME USER;
- ***DELETE*** - виконання команди DELETE;
- ***DROP*** - виконання команд DROP DATABASE і DROP TABLE;

- ***FILE*** - читання та створення файлів на сервері командами SELECT…INTO OUTFILE і LOAD DATA INFILE;
- ***INDEX*** - виконання команд CREATE INDEX і DROP INDEX;
- ***INSERT*** - виконання команди INSERT;
- ***SELECT*** - виконання команди SELECT;
- ***LOCK TABLES*** - виконання команди LOCK TABLES при наявності привілеї SELECT для заблокованих таблиць;
- ***SHOW DATABASES*** - дозволяє відображення всіх баз даних командою SHOW DATABASES;
- ***SUPER*** - привілей адміністратора сервера; зокрема, дозволяє виконання команди SET GLOBAL;
- ***UPDATE*** - дозволяє виконання команди UPDATE;
- ***GRANT OPTION*** - дозволяє призначати і

Областю дії привілеї можуть бути:

- всі бази даних (привілеї називаються глобальними);
- окрема база даних;
- таблиця;
- стовпець таблиці.

Кожен тип привілеї має допустимі області дії. Так, привілеї ***FILE, SHOW DATABASES, RELOAD, SUPER і CREATE USER*** можуть бути тільки глобальними. Привілей ***LOCK TABLES*** застосовується глобально або до окремих баз даних, але не до окремих таблиць. До окремих стовпців таблиці застосовуються тільки привілеї ***SELECT, INSERT і UPDATE.***

2.2 Надання привілеїв

Привілеї користувачам надаються командою

GRANT <Тип привілеї > [(<Список стовпців>)] ON <Область дії > TO <Ідентифікатор користувача>

[WITH GRANT OPTION];

В якості області дії можна вказати одне зі значень:

- ***.** * - привілей діє глобально;
- ***<Ім'я бази даних>.**** - привілей діє для вказаної бази даних;
- * - привілей діє для поточної бази даних;
- ***<Ім'я бази даних>.<Ім'я таблиці>*** або ***<Ім'я таблиці>*** - привілей діє для вказаної таблиці. Якщо потрібно створити привілеї тільки для окремих стовпців, слід перерахувати ці стовпці в дужках перед ключовим словом ON.

Приклади.

- ***GRANT CREATE ON *.* TO ‘anna’@'localhost';***

Команда надає користувачеві ***anna'@’localhost*** привілей на створення баз даних і таблиць в будь-якій базі даних.

- ***GRANT DROP ON SalesDept.* TO ‘anna’@’localhost';***

Команда надає користувачеві ***anna'@'localhost*** привілей на видалення таблиць в базі даних ***SalesDept***, а також на видалення самої бази даних ***SalesDept***.

- ***GRANT SELECT ON SalesDept.Products TO ‘anna’@'localhost';***

Команда надає користувачеві ***anna'@’localhost*** привілей на отримання даних з таблиці ***Products*** бази даних ***SalesDept***.

- ***GRANT UPDATE (price) ON SalesDept.Products TO ‘anna‘@'localhost';***

Команда надає користувачеві ***anna'@’localhost*** привілей на зміну даних в стовпці ***price*** таблиці ***Products***.

2.3 Відміна привілеїв

Для видалення привілеїв використовується команда

REVOKE <ип привілеї > [(<Список стовпців>)] ON <Область дії >

FROM < Ідентифікатор користувача >;

Наприклад:

- ***REVOKE CREATE ON *.* FROM 'anna'@'localhost';***

Команда скасовує глобальний привілей користувача ***'anna'@'localhost'***, котрий дозволяв створення баз даних і таблиць.

- ***REVOKE DROP ON SalesDept.* FROM 'anna'@'localhost';***

Команда скасовує привілей користувача ***'anna'@'localhost'*** на видалення бази даних ***SalesDept*** і таблиць в цій базі даних.

- ***REVOKE SELECT ON SalesDept.Products***

  FROM 'anna'@'localhost';

Команда скасовує привілей користувача ***'anna'@ 'localhost'*** на отримання даних з таблиці ***Products*** бази даних ***SalesDept***.

- ***REVOKE UPDATE (price) ON SalesDept.Products***

  FROM 'anna'@'localhost';

Команда скасовує привілей користувача ***'anna'@ 'localhost'*** на зміну даних в стовпці ***price*** таблиці ***Products***.

2.4 Перегляд привілеїв

Відомості про привілеї доступу містяться в таблицях системної бази даних ***mysql***.

- Глобальні привілеї зберігаються в таблиці ***user*** (користувач). Кожному типу привілеїв відповідає окремий стовпець, що допускає значення '***Y***' (операція дозволена) і '***N***' (операція не дозволена).
- Привілеї, областю дії яких є окрема база даних, зберігаються в таблиці ***db*** (база даних). Кожному привілею відповідає окремий стовпець, можливими значеннями котрого є 'Y' і 'N'.
- Привілеї окремих таблиць, зберігаються в таблиці ***tables_ priv***. Кожен рядок таблиці ***tables_priv*** визначає привілеї доступу конкретного користувача до конкретної таблиці.
- Привілеї для окремих стовпців зберігаються в таблиці ***columns_priv***.

3. Резервування бази даних

3.1.Повне резервування

Для повного резервного копіювання баз даних призначена утиліта ***mysqldump***.

Для запуску утиліти з командного рядка Windows слід виконати команду

mysqldump -u <Ім'я користувача> -p

[Опціональні параметри]

< Копійовані бази даних і таблиці>

> < Шлях та ім'я результуючого файлу>

Після появи запрошення ***Enter password*** (Введіть пароль) введіть пароль користувача.

Вибір опціональних параметрів утиліти залежить від типу резервуються таблиць.

- Для резервування таблиць InnoDB слід вказати параметр-***single -transaction.***
- При резервуванні таблиць MyISAM слід заборонити користувачам зміну даних, щоб уникнути їх неузгодженості. Для цього необхідно вказати параметр ***-lock -all -tables*** або ***-lock -tables***.

Копійовані об'єкти можна задати одним із способів:

- ***-all –databases***

необхідне копіювання всіх баз даних з сервера MySQL.

- ***-databases <Ім'я бази даних1> <Ім'я бази даних2>...*** необхідно скопіювати перераховані бази даних.

- ***<Ім'я бази даних> <Ім'я таблиці1> <Ім'я таблиці2>...*** необхідно скопіювати перераховані таблиці вказаної бази даних.

  Наприклад, команда

  mysqldump -u root -p -single -transaction -flush -logs -Databases SalesDept FinanceDept > "C:\data\full_backup.sql"

виконує резервне копіювання баз даних ***SalesDept*** і ***FinanceDept*** у файл ***full_backup.sql***, що знаходиться в папці C:\data, а команда

mysqldump -u root -p -lock -tables -flush -logs mysql user db tables_priv columns_priv > "C:\data\users.sql"

виконує резервне копіювання таблиць ***user, db, tables_priv*** і ***columns_priv*** системної бази даних ***mysql*** в файл ***users.sql***, що знаходиться в папці C:\data. Таблиці в базі даних ***mysql*** мають тип MyISAM, тому при резервуванні ми вказали параметр***-lock -tables.***

3.1 Відновлення даних

Щоб відновити базу даних з файлу, що містить повну резервну копію, слід виконати команду з командного рядка Windows

mysql -u root -p [<Ім'я бази даних>] < <Шлях та ім'я файлу>

Після появи запрошення Enter password ввести пароль користувача root.

Якщо резервна копія була створена командою

mysqldump -u root -p -single -transaction -flush -logs

-Databases SalesDept FinanceDept > " C: \data full_backup.sql"

то відновити бази даних SalesDept і FinanceDept можна командою

mysql -u root -p < "C:\data full_backup.sql"

Якщо резервна копія була створена за допомогою команди

mysqldump -u root -p -lock -tables -flush -logs

mysql user db tables_priv columns_priv > "C:\data \users.sql"

то при відновленні необхідно вказати ім'я бази даних, в яку будуть поміщені відтворені таблиці user, db, tables_priv і columns_priv:

mysql -u root -p mysql < "C:\data \users.sql"

4. Профілактична перевірка та відновлення таблиць

Для перевірки таблиць виконати команду

CHECK TABLE <Список таблиць>;

Команда CHECK TABLE відображає результат перевірки таблиць. Наприклад, щоб отримати інформацію про стан таблиць db і user системної бази даних mysql, виконайте команду

CHECK TABLE mysql.db, mysql.user;

Якщо в стовпці ***Msg_text*** (текст повідомлення) міститься значення, відмінне від ***OK*** або ***Table is already up to date*** (Таблиця вже перевірена), то таблиця пошкоджена.

Для відновлення таблиці слід виконати такі дії.

1. Вибрати команду ***REPAIR TABLE <Ім'я таблиці> QUICK;***

Результат виконання команди REPAIR TABLE аналогічний результату виконання команди CHECK TABLE. Якщо в останньому рядку в стовпці Msg_text (текст повідомлення) вказано значення OK, то таблиця успішно відновлена. В іншому випадку перейдіть до наступного пункту.

2. Скопіювати файл ***<Ім'я таблиці>. MYD*** з папки ***<Коренева папка MySQL >\data\<Ім'я бази даних>*** в будь-яку резервну папку, тому щоспроби відновлення можуть пошкодити дані, які містяться в цьому файлі.

3. Виконати команду

REPAIR TABLE <Ім'я таблиці>;

Якщо і ця команда не допомогла відновити таблицю, виконати команду

REPAIR TABLE <Ім'я таблиці> EXTENDED;

Якщо знову не вийшло виправити ушкодження, виконати команду

REPAIR TABLE <Ім'я таблиці> USE_FRM;

Параметр ***USE_FRM*** повинен використовуватися тільки в тому випадку, якщо попередні дії не дали потрібного результату.

Якщо таблиця так і не була відновлена, перейти до наступного пункту.

4. Відкрити файл з повною резервною копією бази даних. Знайти у ньому SQL - команду CREATE TABLE для тієї таблиці, яку потрібно відновити. За допомогою цієї команди створити точно таку ж таблицю в іншій базі даних. Потім перемістити файли ***<Ім'я таблиці>. MYI*** і ***<Ім'я таблиці>.frm*** з папки <Коренева папка MySQL > \data\<Ім'я іншої бази даних> в папку ***<Коренева папка MySQL > \data\<Ім'я вихідної бази даних>.*** Повторити дії, описані в п. 3.