

Conversación con Gemini

Exported: 15/6/2026 9:14:17

Jordi H.

Has dicho he hecho un downgrade de mysql 9 a mysql 8 todos los datos los he guardado con un dump y luego los he restaurado. ahora estoy usando dbeaver para mirar cómo ha ido y los datos están bien (eso creo) pero la información de los usuarios y sus grant es inconsistente. creo que no ha ido bien el downgrade. de hecho, me da error el programa que hace login. Me puedes ayudar

Gemini

Es muy probable que tu diagnóstico sea correcto: el downgrade de MySQL 9 a 8.0 suele ser problemático con los usuarios porque MySQL 9 introduce cambios en el diccionario de datos y, sobre todo, porque los "dumps" estándar (como mysqldump) a menudo **no incluyen las tablas de sistema** de la base de datos mysql, que es donde viven los usuarios y sus permisos. Además, MySQL 9 puede utilizar plugins de autenticación o estructuras de tablas de grants que MySQL 8 no reconoce. Aquí tienes los pasos para limpiar la inconsistencia y recuperar el acceso

1. El problema del "Dump"

Si hiciste un mysqldump --all-databases, el volcado intentó sobrescribir la base de datos mysql de la versión 8 con los datos de la versión 9. Esto corrompe las tablas de permisos porque el esquema interno ha cambiado. **Solución recomendada:** No intentes "arreglar" las tablas del sistema. Es mejor recrear los usuarios manualmente o extraer solo sus sentencias de creación

2. Cómo recuperar el acceso (Modo Rescate)

Si el programa de login falla y tú no puedes gestionar permisos, reinicia MySQL saltándote las tablas de privilegios

1. Detén el servicio de MySQL.
2. Inicia MySQL manualmente con el comando: `mysqld_safe --skip-grant-tables &`
3. Entra sin contraseña: `mysql -u root`
4. Ejecuta `FLUSH PRIVILEGES;` (esto es vital para que te deje usar comandos de usuario después).

3. Sincronización de Usuarios y Grants

Para que DBeaver y tu programa vuelvan a funcionar, te sugiero recrear los usuarios específicos de tu aplicación. En MySQL 8, asegúrate de usar el plugin de autenticación compatible: `mysql_native_password`

CODE

```
-- Cambia 'usuario' y 'password' por los de tu app
CREATE USER 'mi_app_user'@'localhost' IDENTIFIED WITH mysql_native_password BY 'tu_password';

-- Asigna los permisos a tu base de datos restaurada
GRANT ALL PRIVILEGES ON tu_base_datos.* TO 'mi_app_user'@'localhost';

FLUSH PRIVILEGES;
```

Nota importante: MySQL 8 introdujo `caching_sha2_password` como predeterminado. Si tu programa es antiguo o usas conectores de Python que no están actualizados, usa `mysql_native_password` como en el ejemplo d

e arriba para evitar errores de "Authentication plugin not supported"

4. ¿Por qué DBeaver dice que "están bien"?

DBeaver suele usar una conexión con un usuario que sí tiene permisos (quizás el root que configuraste al instalar la v8), pero tu aplicación probablemente usa un usuario distinto que se quedó en el "limbo" durante el proceso de dump/restore. **Checklist de verificación**

- Ejecuta `SELECT User, Host, plugin FROM mysql.user;` y verifica si los usuarios tienen el plugin correcto.
- Si ves usuarios que vienen de la v9 con plugins extraños, bórralos (`DROP USER`) y vuélvelos a crear.

¿Cómo lo tienes montado?

Si estás usando Python para conectar con la base de datos, ¿podrías decirme qué librería usas (mysql-connector, PyMySQL o SQLAlchemy)? Dependiendo de eso, el error de login podría solucionarse simplemente ajustando el plugin de autenticación en la sentencia `CREATE USER`. ¿Te gustaría que te ayude a generar un script de Python para verificar rápidamente si la conexión desde el código vuelve a ser éxito?

Jordi H.

Has dicho tengo esto `mysql> SELECT User, Host, plugin FROM mysql.user;`

```
+-----+-----+-----+ | User      | Host    | plugin
| +-----+-----+-----+ | root     | %       | caching_sha2_password |
| dplus   | 10.88.0.1 | caching_sha2_password | | mysql.infoschema | localhos
t | caching_sha2_password | | mysql.session   | localhost | caching_sha2_password | | mysql.sys
  | localhost | caching_sha2_password | | root           | localhost | caching_sha2_passw
```

ord | +-----+-----+-----+ dame la sentencia para cambiar el plugin del usuario dplus y hago la

Gemini

El problema es que `caching_sha2_password` es el método de cifrado moderno, pero muchas librerías de Python (como las versiones estándar de `mysql-connector-python` o `PyMySQL`) a veces fallan si no tienen configurado el soporte para SHA2 o si el servidor no tiene los certificados SSL configurados correctamente. Para hacer la prueba de compatibilidad máxima, cambia el usuario `dplus` al plugin antiguo (`mysql_native_password`

Sentencia SQL

Ejecuta esto en tu terminal de MySQL:SQL

CODE

```
ALTER USER 'dplus'@'10.88.0.1' IDENTIFIED WITH mysql_native_password BY 'tu_password_aqui';

-- No olvides refrescar los privilegios para que el cambio sea inmediato
FLUSH PRIVILEGES;
```

¿Por qué esto suele arreglar el login en Python?

Cuando usas Python para analítica de datos, la cadena de conexión a veces no sabe cómo gestionar el "handshake" del nuevo plugin de MySQL 8/9. Al volver a `mysql_native_password`, usas un método que cualquier conector entiende sin configuraciones extra. **Si después de esto sigue fallando, fíjate en dos cosas**

1. El Host: El usuario está configurado para entrar desde la IP 10.88.0.1. Si tu programa corre desde otra IP

(o desde el mismo servidor pero detectado como localhost), el error será de "Access Denied"

2. Librería Python: Si usas pandas con sqlalchemy, asegúrate de tener instalado cryptography en tu entorno de Python (pip install cryptography), ya que es necesario para que Python entienda el plugin caching_sha2_password

¿Te ha dado algún mensaje de error específico el programa al intentar conectar después de este cambio?

Jordi H.

Has dicho mysql> ALTER USER 'dplus'@'10.88.0.1' IDENTIFIED WITH mysql_native_password BY 'discoveryplus'; ERROR 1524 (HY000): Plugin 'mysql_native_password' is not loaded

Gemini

Este es un síntoma clásico del salto de **MySQL 9** a **MySQL 8.0.34+**. En las versiones más recientes, el equipo de Oracle ha empezado a retirar mysql_native_password. En MySQL 9 viene desactivado por defecto o incluso eliminado, y si tu instalación de MySQL 8 es muy reciente (como la 8.4 LTS o las últimas de la rama 8.0), el plugin está ahí pero **no está cargado**. Tienes dos opciones para solucionar es

Opción A: Cargar el plugin en MySQL 8 (Recomendado)

Para que MySQL reconozca ese método de autenticación, debes habilitarlo en el archivo de configuración del servidor (generalmente my.cnf o my.ini)

1. Busca tu archivo de configuración (en Linux suele estar en /etc/mysql/my.cnf).
2. Añade o edita la sección [mysqld] con esta línea: Ini, TOML

CODE

```
[mysqld]
mysql_native_password=ON
```

3. Reinicia el servicio de MySQL para que los cambios surtan efecto.
4. Una vez reiniciado, vuelve a lanzar el comando ALTER USER que intentaste antes.

Opción B: Usar el plugin moderno en Python

Si no quieres tocar la configuración del servidor, el problema es que tu script de Python no sabe hablar con caching_sha2_password. Para arreglarlo desde el lado de Python sin cambiar el plugin en la base de datos

1. Asegúrate de tener instalada la librería de criptografía (es la que permite a Python manejar el nuevo plugin): Bas

CODE

```
pip install cryptography
```

2. Si usas SQLAlchemy o un conector directo, añade el parámetro allow_native_password=True o asegúrate de que la cadena de conexión sea compatible

¿Por qué te da el error 1524?

El servidor simplemente no tiene la "pieza" de software necesaria para entender ese método de contraseña antigua porque se considera menos seguro. **¿Qué versión exacta de MySQL 8 estás usando ahora?** (Puedes verlo con SELECT VERSION();). Si te has bajado la **8.4 LTS**, el plugin mysql_native_password es oficialmente "deprecado" y requiere ese paso extra en el archivo de configuración para funcionar. ¿Quieres

que te ayude a localizar el archivo de configuración en tu sist

Jordi H.

Has dicho creo que el problema no es el plugin de password sino que la tabla de grants no es compatible.
dame el comando para hacer grant de todo sobre la base de datos discoveryplus_develop para el usuario dplu

Gemini

Tienes razón en tu sospecha. Si el dump de MySQL 9 sobrescribió la base de datos mysql de la versión 8, la estructura interna de la tabla mysql.db o mysql.tables_priv puede estar corrupta o tener columnas que l a versión 8 no sabe interpretar, lo que rompe la lógica de los **Grants**. Para solucionar esto y asignar todos l os permisos a tu base de datos de desarrollo, usa esta sentenc

Comando de Grant Total

Ejecuta esto desde un terminal con privilegios de root:SQL

CODE

```
GRANT ALL PRIVILEGES ON discoveryplus_develop.* TO 'dplus'@'10.88.0.1';

-- Fundamental para que MySQL recargue las tablas de memoria
FLUSH PRIVILEGES;
```

Si el comando anterior te da error (Escenario de Inconsistencia)

Si al ejecutar el GRANT recibes un error tipo "Table 'mysql.user' doesn't exist" o algo sobre columnas desconocidas, significa que el diccionario de datos está realmente dañado por el downgrade. En ese caso, l a solución "limpia" e

1. Borrar el usuario problemático:SQL

CODE

```
DROP USER 'dplus'@'10.88.0.1';
```

2. Recrearlo de cero (esto insertará los registros con la estructura correcta de la v8):SQL

CODE

```
CREATE USER 'dplus'@'10.88.0.1' IDENTIFIED BY 'discoveryplus';
GRANT ALL PRIVILEGES ON discoveryplus_develop.* TO 'dplus'@'10.88.0.1';
FLUSH PRIVILEGES;
```

¿Por qué ocurre esto en un Downgrade?

Cuando pasas de una versión superior a una inferior, el motor de base de datos se encuentra con metadatos que no comprende

Un último consejo para tu entorno de Python

Como mencionaste que te vas a centrar en **analítica de datos**, si después de arreglar los permisos sigues teniendo problemas de conexión desde tus scripts, verifica si estás usando mysql-connector-python. A veces , simplemente actualizar la librería en tu entorno virtual resuelve conflictos de protocolo:Ba

CODE

```
pip install --upgrade mysql-connector-python cryptography
```

¿Te ha funcionado el GRANT o te ha devuelto algún código de error específico? Si te sigue fallando, podemos intentar reparar las tablas de sistema con `mysql_upgrade` (aunque en versiones 8.0+ esto suele ejecutarse solo al arrancar

Jordi H.

Has dicho ha funcionado. y veo los grants desde dbeaver que es la prueba definitiva

Gemini

¡Excelente noticia! Si ya puedes ver los **Grants** correctamente en el inspector de DBeaver, significa que el diccionario de datos ha recuperado la coherencia y el usuario `dplus` ya tiene su "hoja de ruta" de permisos clara. Esa es la **prueba de fuego** porque DBeaver consulta directamente las tablas de metadatos (`information_schema` y `mysql`), así que si él lo ve, el motor de base de datos ya sabe qué hacer con las peticiones que vengan de tu progr

Un par de consejos para tu flujo de Analítica con Python:

Como mencionaste que te vas a centrar en **Python y Análisis de Datos**, ahora que la base de datos está estable, te sugiero tener en cuenta estos dos puntos para evitar que el login te vuelva a dar guerra

1. Librería de conexión: Si usas Pandas, lo más común es usar SQLAlchemy. Asegúrate de que tu cadena de conexión tenga este formato para evitar líos de protocolos
`mysql+mysqlconnector://dplus:discoveryplus@10.88.0.1/discoveryplus_develop`
2. Entorno virtual: Mantén siempre `cryptography` instalado en tu entorno de trabajo. Aunque ahora te funcione, si en algún momento decides volver a `caching_sha2_password` (el estándar de MySQL 8), Python lo necesitará obligatoriamente para cifrar la clave durante el handshake

¿Qué sigue ahora?

Ahora que los permisos están en orden, ¿quieres que te ayude a montar un **script rápido en Python** para extraer los primeros datos de `discoveryplus_develop` y cargarlos en un `DataFrame` de Pandas? O si prefieres, podemos revisar si hay alguna otra inconsistencia en los tipos de datos tras el downgrade
