Gestire utenti e permessi con MySQL

Gestire gli utenti

Per aggiungere, modificare o rimuovere un utente ci sono due sistemi:

- Il primo consente di creare, disabilitare e modificare gli utenti, le password e i permessi con una sola query, tramite l'utilizzo dei comandi **GRANT** e **REVOKE**. Questo sistema, però, non permette di rimuoverli, ma solo di disabilitarli.
- Il secondo, sicuramente più lungo e complesso, permette di capire, bene, come MySQL gestisce le informazioni all'interno, creando utenti e permessi *ad hoc*

CONCESSIONE PERMESSI CON GRANT e REVOCA DI PRIVILEGI TRAMITE REVOKE

Un problema molto frequente quando si lavora con database in ambito industriale è dover assegnare **privilegi** o **permessi limitati** a particolari utenti: normalmente è compito del sistemista o DBA (Database Administrator).

MySQL :: MySQL 8.0 Reference

MySQL

Permessi all'utente: livelli

Per utilizzare l'istruzione **GRANT** è quindi necessario elencare i permessi da dare all'utente, e questi possono essere:

- a livello **GLOBALE**, quando nella clausola ON viene specificato ***.***
- a livello di **DATABASE**, quando nella clausola ON viene impostato **nome_database.***
- a livello di **TABELLA**; quando viene utilizzato nella clausola ON **nome_database.nome_tabella**
- a livello di **Colonna** (si applicano alle singole colonne di una tabella)
- a livello di **Routine** (si applicano alle *stored routines: funzioni e procedure*)

online

MySQL

MySQL è molto flessibile a riguardo e ci viene incontro tramite gli statement **GRANT** e **REVOKE**.

Il comando **GRANT** permette allo stesso tempo di creare un utente e di assegnargli dei permessi specifici. Vediamone la sintassi:

```
GRANT
<istruzioni_consentite>
ON <database>.<tabella>
TO <utente>@<host>
IDENTIFIED BY <password>;
```

Grant: la sintassi

- **istruzioni_consentite**: è una lista di *statements* SQL che si vogliono permettere all'utente (CREATE, SELECT, UPDATE, DELETE, ALTER, EXECUTE, ecc..). Se si vogliono dare all'utente permessi completi si può utilizzare la parola chiave ALL.

- **database**: è il nome del database che stiamo prendendo in considerazione.

- **tabella**: inserendo il nome di una tabella, si fa riferimento solo ad essa. Per tutte le altre tabelle non varranno le regole che stiamo specificando. Se si vuole fare riferimento a tutte le tabelle si può utilizzare il carattere asterisco (*).

- **utente**: specifica il nome dell'utente che vogliamo creare

- **host**: specifica il/gli host da cui è ammessa la connessione

- **password**: specifica la password associata all'utente che stiamo creando. La password va scritta "in chiaro". Se si desidera inserire la password in forma criptata tramite la **funzione PASSWORD()** di MySQL, si deve far precedere la stringa criptata dalla parola PASSWORD.

Esempio (1)

Immaginiamo di stare gestendo un database per un **Cinema**.

Immaginiamo che questo database si chiami "Cinema".

Immaginiamo infine che questo cinema voglia che gli **accessi** al database avvengano solo da **utenti** di questi **tre tipi**:

- **Root**: questo è l'utente a cui **tutto è permesso**. L'utente root può leggere e modificare i dati delle tabelle; inserire nuove tuple; creare e cancellare tabelle e alterare la struttura di quelle già esistenti. Insomma, può fare tutto.
- **Amministratori**: questi utenti possono inserire, leggere, modificare e cancellare i dati (record) delle tabelle; possono impostare References**. Non possono però creare** nuove tabelle **o alterare la struttura** di tabelle esistenti (Alter, Drop).
- **Statisti**: questo tipo di utenti ha **accesso in lettura** a tutto al database ma non ha in nessun caso i permessi di scrittura. Questi utenti possono solo leggere le tuple delle varie tabelle, senza alterarle in nessun modo. Possono, insomma, solo utilizzare i dati per produrre delle statistiche.

Esempio (2)

```
GRANT SELECT
ON Cinema.*
TO 'cinema_stat'
IDENTIFIED BY 'statista_pass';

GRANT SELECT, INSERT, UPDATE, DELETE
ON Cinema.*
TO 'cinema_admin'
IDENTIFIED BY 'admin_pass';

GRANT ALL
ON Cinema.*
TO 'cinema_root@localhost'
IDENTIFIED BY PASSWORD '*6C8989366EA95CEF4';
```

utente root:
due particolarità:
controllo di sicurezza e
password non in chiaro

dall' utente con più restrizioni
*(**statisti**)*

per arrivare a quello più libero
*(**root**):*

In tutti e tre i casi si è utilizzata la sintassi **Cinema.*** per indicare che i permessi verranno applicati a **tutte le tabelle** del database

Utente root: due particolarità

Notiamo infine due particolarità utilizzate per l'utente root:

- Il suffisso **"@localhost"** (con la sintassi come da esempio) è un **controllo di sicurezza** che specifica che le connessioni dall'utente root possono essere accettate solo se provenienti da localhost (la macchina locale). E' possibile specificare anche uno specifico IP o un range di IP.
- La **password** dell'utente root non viene scritta in chiaro ma come **stringa hash** restituita dalla funzione **PASSWORD()** di MySQL.

Revoke

L'istruzione **REVOKE** svolge la funzione opposta a GRANT, e cioè **rimuovere permessi**. Ne vediamo solo la sintassi in quanto molto simile all'istruzione GRANT:

REVOKE <istruzioni_revocate>
ON <database>.<tabella>
FROM <utente>;

valgono le stesse regole sopra viste per GRANT.

Sitografia

***Dispensa della docente** con cenno ad autorizzazioni con MS SQL Server*

(tratta da risorsa online o tutorial)

Manuale MySQL – References

Articolo “… da command line” (Linux)

Articolo da HTML.it

Guida online con approfondimento sui Livelli