Programmazione Web con PHP e SQL

Gestione degli utenti in SQL (parte 2)

prof. Leonardo Essam Dei Rossi

ITT "M. Buonarroti" - Trento (TN)

Anno scolastico 2025/2026

Indice

Indice

I privilegi amministrativi (1)

I privilegi amministrativi in MySQL sono permessi di alto livello che consentono a un utente di eseguire operazioni che influenzano il funzionamento dell'intero server database, anziché limitarsi alla sola gestione dei dati all'interno di specifici database o tabelle.

A differenza dei privilegi "sugli oggetti" (come `SELECT`, `INSERT`, o `UPDATE` che agiscono su tabelle specifiche), i privilegi amministrativi operano a livello globale.

I privilegi amministrativi (1)

I privilegi amministrativi in MySQL sono permessi di alto livello che consentono a un utente di eseguire operazioni che influenzano il funzionamento dell'intero server database, anziché limitarsi alla sola gestione dei dati all'interno di specifici database o tabelle.

A differenza dei privilegi "sugli oggetti" (come `SELECT`, `INSERT`, o `UPDATE` che agiscono su tabelle specifiche), i privilegi amministrativi operano a livello globale.

I privilegi amministrativi (2)

- `CREATE USER`: Consente di creare, modificare, rinominare e rimuovere account utente dal server;
- `PROCESS`: Consente di visualizzare le informazioni su tutti i thread (le query e le connessioni) attualmente in esecuzione sul server (ad esempio, usando il comando `SHOW PROCESSLIST`);

I privilegi amministrativi (2)

- `CREATE USER`: Consente di creare, modificare, rinominare e rimuovere account utente dal server;
- `PROCESS`: Consente di visualizzare le informazioni su tutti i thread (le query e le connessioni) attualmente in esecuzione sul server (ad esempio, usando il comando `SHOW PROCESSLIST`);

I privilegi amministrativi (3)

- `RELOAD`: Permette di utilizzare i comandi FLUSH per ricaricare le tabelle di sistema, svuotare le cache o ruotare i file di log;
- `SHUTDOWN`: Consente di arrestare il server MySQL;
- `SUPER`: Storicamente, era un privilegio "onnipotente" per gli amministratori. Permette di terminare le connessioni di altri utenti, modificare le variabili di configurazione globali a runtime e bypassare i limiti di connessione.

Nota: Nelle versioni più recenti di MySQL, `SUPER` sta venendo progressivamente deprecato e suddiviso in privilegi dinamici più specifici per migliorare la sicurezza.

I privilegi amministrativi (3)

- `RELOAD`: Permette di utilizzare i comandi FLUSH per ricaricare le tabelle di sistema, svuotare le cache o ruotare i file di log;
- `SHUTDOWN`: Consente di arrestare il server MySQL;
- `SUPER`: Storicamente, era un privilegio "onnipotente" per gli amministratori. Permette di terminare le connessioni di altri utenti, modificare le variabili di configurazione globali a runtime e bypassare i limiti di connessione.

Nota: Nelle versioni più recenti di MySQL, `SUPER` sta venendo progressivamente deprecato e suddiviso in privilegi dinamici più specifici per migliorare la sicurezza.

I privilegi amministrativi (3)

- `RELOAD`: Permette di utilizzare i comandi FLUSH per ricaricare le tabelle di sistema, svuotare le cache o ruotare i file di log;
- `SHUTDOWN`: Consente di arrestare il server MySQL;
- `SUPER`: Storicamente, era un privilegio "onnipotente" per gli amministratori. Permette di terminare le connessioni di altri utenti, modificare le variabili di configurazione globali a runtime e bypassare i limiti di connessione.

Nota: Nelle versioni più recenti di MySQL, `SUPER` sta venendo progressivamente deprecato e suddiviso in privilegi dinamici più specifici per migliorare la sicurezza.

Assegnazione dei privilegi

I privilegi amministrativi devono essere assegnati a **livello globale**. Nella sintassi SQL, questo viene rappresentato con `*.*` (che significa "su tutti i database e tutte le tabelle").

Un esempio di comando per concedere un privilegio amministrativo è:

```
GRANT RELOAD, PROCESS ON *.* TO 'nome_utente'@'localhost';
```

Assegnazione dei privilegi

I privilegi amministrativi devono essere assegnati a **livello globale**. Nella sintassi SQL, questo viene rappresentato con `*.*` (che significa "su tutti i database e tutte le tabelle").

Un esempio di comando per concedere un privilegio amministrativo è:

```
GRANT RELOAD, PROCESS ON *.* TO 'nome_utente'@'localhost';
```

Best-practice di sicurezza

Quando si gestiscono questi permessi, è fondamentale applicare il **Principio del Privilegio Minimo (PoLP)**[1].

Ciò significa concedere agli utenti solo ed esclusivamente i privilegi strettamente necessari per svolgere il loro lavoro. Privilegi come `FILE`, `SHUTDOWN` o `SUPER` dovrebbero essere riservati unicamente agli amministratori di database (DBA) fidati.

[1] Dall'inglese: *Principle of Least Privileges*

Indice

I privilegi sulla struttura (1)

I privilegi di struttura in MySQL sono permessi che consentono a un utente di creare, modificare o eliminare l'architettura logica del database, ovvero i suoi "contenitori" e le sue regole.

Si differenziano nettamente sia dai privilegi sui dati (come `SELECT`, `INSERT` o `UPDATE`, che manipolano le informazioni dentro le tabelle), sia dai privilegi amministrativi (che gestiscono l'intero server). I privilegi di struttura agiscono sullo scheletro del database.

Tratto da: [1]

I privilegi sulla struttura (2)

- `CREATE`: Permette di creare nuovi database o nuove tabelle;
- `ALTER`: Consente di modificare la struttura di una tabella esistente (ad esempio, aggiungere o rimuovere una colonna, cambiare il tipo di dato di un campo o rinominare la tabella);
- `DROP`: È il permesso di distruzione. Consente di eliminare in modo permanente interi database, tabelle o viste. **Attenzione:** l'uso improprio di questo privilegio causa la perdita irreparabile della struttura e dei dati in essa contenuti.

I privilegi sulla struttura (2)

- `CREATE`: Permette di creare nuovi database o nuove tabelle;
- `ALTER`: Consente di modificare la struttura di una tabella esistente (ad esempio, aggiungere o rimuovere una colonna, cambiare il tipo di dato di un campo o rinominare la tabella);
- `DROP`: È il permesso di distruzione. Consente di eliminare in modo permanente interi database, tabelle o viste. **Attenzione:** l'uso improprio di questo privilegio causa la perdita irreparabile della struttura e dei dati in essa contenuti.

I privilegi sulla struttura (2)

- `CREATE`: Permette di creare nuovi database o nuove tabelle;
- `ALTER`: Consente di modificare la struttura di una tabella esistente (ad esempio, aggiungere o rimuovere una colonna, cambiare il tipo di dato di un campo o rinominare la tabella);
- `DROP`: È il permesso di distruzione. Consente di eliminare in modo permanente interi database, tabelle o viste. **Attenzione:** l'uso improprio di questo privilegio causa la perdita irreparabile della struttura e dei dati in essa contenuti.

I privilegi sulla struttura (3)

- `INDEX`: Permette di creare o rimuovere indici su una tabella. Gli indici non alterano i dati, ma strutturano il modo in cui il motore del database li cerca, rendendo le query molto più veloci;
- `CREATE VIEW` e `SHOW VIEW`: Consentono, rispettivamente, di creare viste (tabelle "virtuali" basate su query salvate) e di ispezionarne la query sottostante;
- `CREATE ROUTINE` e `ALTER ROUTINE`: Riguardano la gestione della logica programmata direttamente nel database, permettendo di creare, modificare o eliminare stored procedure (procedure archiviate) e funzioni;

I privilegi sulla struttura (3)

- `INDEX`: Permette di creare o rimuovere indici su una tabella. Gli indici non alterano i dati, ma strutturano il modo in cui il motore del database li cerca, rendendo le query molto più veloci;
- `CREATE VIEW` e `SHOW VIEW`: Consentono, rispettivamente, di creare viste (tabelle "virtuali" basate su query salvate) e di ispezionarne la query sottostante;
- `CREATE ROUTINE` e `ALTER ROUTINE`: Riguardano la gestione della logica programmata direttamente nel database, permettendo di creare, modificare o eliminare stored procedure (procedure archiviate) e funzioni;

I privilegi sulla struttura (3)

- `INDEX`: Permette di creare o rimuovere indici su una tabella. Gli indici non alterano i dati, ma strutturano il modo in cui il motore del database li cerca, rendendo le query molto più veloci;

- `CREATE VIEW` e `SHOW VIEW`: Consentono, rispettivamente, di creare viste (tabelle "virtuali" basate su query salvate) e di ispezionarne la query sottostante;

- `CREATE ROUTINE` e `ALTER ROUTINE`: Riguardano la gestione della logica programmata direttamente nel database, permettendo di creare, modificare o eliminare stored procedure (procedure archiviate) e funzioni;

I privilegi sulla struttura (4)

- `TRIGGER`: Consente di creare o rimuovere trigger, ovvero regole strutturali che scatenano azioni automatiche quando si verifica un determinato evento (come un inserimento o una cancellazione) in una tabella;

- `EVENT`: Permette di creare, modificare, eliminare o visualizzare eventi pianificati (task che il database esegue autonomamente a intervalli regolari).

I privilegi sulla struttura (4)

- `TRIGGER`: Consente di creare o rimuovere trigger, ovvero regole strutturali che scatenano azioni automatiche quando si verifica un determinato evento (come un inserimento o una cancellazione) in una tabella;

- `EVENT`: Permette di creare, modificare, eliminare o visualizzare eventi pianificati (task che il database esegue autonomamente a intervalli regolari).

Assegnazione dei privilegi (1)

A differenza dei privilegi amministrativi che sono sempre globali, i privilegi di struttura sono molto più flessibili e possono essere assegnati a diversi livelli, a seconda di quanto controllo vuoi concedere:

- **Livello Globale** (`*.*`)**:** L'utente può creare o distruggere strutture in qualsiasi database sul server;
- **Livello Database** (`nome_database.*`)**:** L'utente può gestire le strutture solo all'interno di uno specifico database. Questo è il livello più comune per uno sviluppatore;
- **Livello Tabella** (`nome_database.nome_tabella`)**:** L'utente può alterare o creare indici solo per una specifica tabella.

Assegnazione dei privilegi (1)

A differenza dei privilegi amministrativi che sono sempre globali, i privilegi di struttura sono molto più flessibili e possono essere assegnati a diversi livelli, a seconda di quanto controllo vuoi concedere:

- **Livello Globale** (`*.*`)**:** L'utente può creare o distruggere strutture in qualsiasi database sul server;
- **Livello Database** (`nome_database.*`)**:** L'utente può gestire le strutture solo all'interno di uno specifico database. Questo è il livello più comune per uno sviluppatore;
- **Livello Tabella** (`nome_database.nome_tabella`)**:** L'utente può alterare o creare indici solo per una specifica tabella.

Assegnazione dei privilegi (1)

A differenza dei privilegi amministrativi che sono sempre globali, i privilegi di struttura sono molto più flessibili e possono essere assegnati a diversi livelli, a seconda di quanto controllo vuoi concedere:

- **Livello Globale** (`*.*`)**:** L'utente può creare o distruggere strutture in qualsiasi database sul server;
- **Livello Database (**`nome_database.*`**):** L'utente può gestire le strutture solo all'interno di uno specifico database. Questo è il livello più comune per uno sviluppatore;
- **Livello Tabella (**`nome_database.nome_tabella`**):** L'utente può alterare o creare indici solo per una specifica tabella.

Assegnazione dei privilegi (1)

A differenza dei privilegi amministrativi che sono sempre globali, i privilegi di struttura sono molto più flessibili e possono essere assegnati a diversi livelli, a seconda di quanto controllo vuoi concedere:

- **Livello Globale** (`*.*`)**:** L'utente può creare o distruggere strutture in qualsiasi database sul server;
- **Livello Database (**`nome_database.*`**):** L'utente può gestire le strutture solo all'interno di uno specifico database. Questo è il livello più comune per uno sviluppatore;
- **Livello Tabella (**`nome_database.nome_tabella`**):** L'utente può alterare o creare indici solo per una specifica tabella.

Assegnazione dei privilegi (2)

Ecco un esempio pratico per concedere a uno sviluppatore il permesso di costruire e modificare la struttura di un database specifico, ma senza il permesso distruttivo di eliminare le tabelle:

```
GRANT CREATE, ALTER, INDEX, CREATE VIEW
ON mio_database.*
TO 'sviluppatore'@'localhost';
```

Best-practice di sicurezza

Negli ambienti di sviluppo, è normale che i programmatori abbiano molti privilegi di struttura per poter costruire l'applicazione.

Tuttavia, negli ambienti di produzione, le regole cambiano: le applicazioni dovrebbero collegarsi al database con utenti che possiedono solo privilegi sui dati (`SELECT`, `INSERT`, `UPDATE`, `DELETE`).

Le modifiche strutturali (come aggiungere una colonna con `ALTER`) dovrebbero essere fatte solo da amministratori o script di migrazione autorizzati, per evitare che un bug o un attacco compromettano l'architettura del database.

Indice

Esercizio #1

Si scrivano le query per:

1. Creare un nuovo utente chiamato `db_architect` che si collega da `localhost` (password: `Costruttore!23`);
2. Assegnare i privilegi di struttura per creare nuovi database e nuove tabelle ovunque nel server, ma senza dargli permessi sui dati (`SELECT`, `INSERT`, ecc.) o permessi di distruzione (`DROP`).

Esercizio #2

Si scrivano le query per:

1. Concedere all'utente esistente `dev_mario@localhost` i permessi strutturali per modificare le tabelle esistenti (aggiungere colonne, ecc.), creare indici e creare viste, ma esclusivamente all'interno del database `webapp_db`.

Indice

Attività: flipped-classroom

Per la prossima lezione, preparare in autonomia una breve spiegazione su come funziona la revoca dei privilegi.

Verranno chiamati $n, n \in \{1, ..., 15\}$ *studenti a esporre alla classe la ricerca svolta.*

Riferimenti e approfondimenti

[1] Atlassian, *How to grant all privileges on a database in MySQL*, [Online; accessed 25-February-2026], 2026. indirizzo: https://www.atlassian.com/data/admin/how-to-grant-all-privileges-on-a-database-in-mysql.