Programmazione Web con PHP e SQL

Gestione degli utenti in SQL (parte 1)

prof. Leonardo Essam Dei Rossi

ITT "M. Buonarroti" - Trento (TN)

Anno scolastico 2025/2026

Indice

1. Introduzione
   - Creazione di un utente
   - Password e sicurezza
   - Esercizio #1
2. Gestione dei privilegi
   - I privilegi in MySQL
   - Privilegi sui dati (Data privileges)
3. Privilegi amministrativi
   - Esercizio #2
4. Esercizio #3

Indice

1. Introduzione
   - Creazione di un utente
   - Password e sicurezza
   - Esercizio #1
2. Gestione dei privilegi
   - I privilegi in MySQL
   - Privilegi sui dati (Data privileges)
3. Privilegi amministrativi
   - Esercizio #2
4. Esercizio #3

Introduzione

La gestione degli utenti è un'operazione fondamentale in un server SQL.

Questo perché per motivi di sicurezza è cosa *buona e giusta* gestire in modo granulare i permessi di accesso ai vari dati contenuti nel DBMS.

MySQL permette di gestire i permessi a livelli diversi:

1. Per singolo utente;
2. [...] + host/indirizzo di connessione;
3. [...] + [...] + per database.

Introduzione

La gestione degli utenti è un'operazione fondamentale in un server SQL.

Questo perché per motivi di sicurezza è cosa *buona e giusta* gestire in modo granulare i permessi di accesso ai vari dati contenuti nel DBMS.

MySQL permette di gestire i permessi a livelli diversi:

1. Per singolo utente;
2. [...] + host/indirizzo di connessione;
3. [...] + [...] + per database.

Introduzione

La gestione degli utenti è un'operazione fondamentale in un server SQL.

Questo perché per motivi di sicurezza è cosa *buona e giusta* gestire in modo granulare i permessi di accesso ai vari dati contenuti nel DBMS.

MySQL permette di gestire i permessi a livelli diversi:

1. Per singolo utente;
2. [...] + host/indirizzo di connessione;
3. [...] + [...] + per database.

Introduzione

La gestione degli utenti è un'operazione fondamentale in un server SQL.

Questo perché per motivi di sicurezza è cosa *buona e giusta* gestire in modo granulare i permessi di accesso ai vari dati contenuti nel DBMS.

MySQL permette di gestire i permessi a livelli diversi:

1. Per singolo utente;
2. [...] + host/indirizzo di connessione;
3. [...] + [...] + per database.

Introduzione

La gestione degli utenti è un'operazione fondamentale in un server SQL.

Questo perché per motivi di sicurezza è cosa *buona e giusta* gestire in modo granulare i permessi di accesso ai vari dati contenuti nel DBMS.

MySQL permette di gestire i permessi a livelli diversi:

1. Per singolo utente;
2. [...] + host/indirizzo di connessione;
3. [...] + [...] + per database.

Creazione di un utente (1)

Analizziamo la seguente query SQL:

```
CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'secure_password';
```

Q: *cosa notiamo dalla query?*

Creazione di un utente (1)

Analizziamo la seguente query SQL:

```
CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'secure_password';
```

Q: *cosa notiamo dalla query?*

Creazione di un utente (2)

La query si compone di 3 (tre) elementi:

1. Il nome dell'utente (`'new_user'`);
2. L'host dal quale l'utente si può collegare (`'localhost'`);
3. La password dell'utente (`'secure_password'`).

Facendo un paragone con PHP, osserviamo che:

1. Il nome utente è lo stesso che va nel parametro `$user` del costruttore della classe `mysqli`;
2. La password, allo stesso modo, è quella che va nel parametro `$password` del costruttore della classe `mysqli`.

Creazione di un utente (2)

La query si compone di 3 (tre) elementi:

1. Il nome dell'utente (`'new_user'`);
2. L'host dal quale l'utente si può collegare (`'localhost'`);
3. La password dell'utente (`'secure_password'`).

Facendo un paragone con PHP, osserviamo che:

1. Il nome utente è lo stesso che va nel parametro `$user` del costruttore della classe `mysqli`;
2. La password, allo stesso modo, è quella che va nel parametro `$password` del costruttore della classe `mysqli`.

Creazione di un utente (3)

Attenzione però che l'host che si inserisce nella query di creazione utente non è lo stesso che va nel costruttore della classe `mysqli`!

Alcuni valori che il campo `host` può assumere:

- `'localhost'`: l'utente si può collegare solo dal server in cui MySQL è in esecuzione;
- `'%'`: l'utente si può collegare da un qualsiasi indirizzo remoto;
- `'192.168.1.50'`: l'utente si può collegare solo dall'indirizzo specificato.

Q: Quali rischi ci sono per la sicurezza?

Creazione di un utente (3)

Attenzione però che l'host che si inserisce nella query di creazione utente non è lo stesso che va nel costruttore della classe `mysqli`!

Alcuni valori che il campo `host` può assumere:

- `'localhost'`: l'utente si può collegare solo dal server in cui MySQL è in esecuzione;
- `'%'`: l'utente si può collegare da un qualsiasi indirizzo remoto;
- `'192.168.1.50'`: l'utente si può collegare solo dall'indirizzo specificato.

Q: Quali rischi ci sono per la sicurezza?

Creazione di un utente (3)

Attenzione però che l'host che si inserisce nella query di creazione utente non è lo stesso che va nel costruttore della classe `mysqli`!

Alcuni valori che il campo `host` può assumere:

- `'localhost'`: l'utente si può collegare solo dal server in cui MySQL è in esecuzione;
- `'%'`: l'utente si può collegare da un qualsiasi indirizzo remoto;
- `'192.168.1.50'`: l'utente si può collegare solo dall'indirizzo specificato.

Q: Quali rischi ci sono per la sicurezza?

Creazione di un utente (3)

Attenzione però che l'host che si inserisce nella query di creazione utente non è lo stesso che va nel costruttore della classe `mysqli`!

Alcuni valori che il campo `host` può assumere:

- `'localhost'`: l'utente si può collegare solo dal server in cui MySQL è in esecuzione;
- `'%'`: l'utente si può collegare da un qualsiasi indirizzo remoto;
- `'192.168.1.50'`: l'utente si può collegare solo dall'indirizzo specificato.

Q: Quali rischi ci sono per la sicurezza?

Creazione di un utente (3)

Attenzione però che l'host che si inserisce nella query di creazione utente non è lo stesso che va nel costruttore della classe `mysqli`!

Alcuni valori che il campo `host` può assumere:

- `'localhost'`: l'utente si può collegare solo dal server in cui MySQL è in esecuzione;
- `'%'`: l'utente si può collegare da un qualsiasi indirizzo remoto;
- `'192.168.1.50'`: l'utente si può collegare solo dall'indirizzo specificato.

Q: Quali rischi ci sono per la sicurezza?

Password e sicurezza (1)

La gestione della password è una delle operazioni più delicate in un server MySQL, o, più in generale, in un server qualsiasi.

In MySQL, l'algoritmo predefinito per la gestione delle password dipende dalla versione che si sta utilizzando. C'è stato un cambiamento significativo tra la versione 5.7 e la 8.0 che ha ridefinito gli standard di sicurezza.

Password e sicurezza (2)

MySQL 8.0 e successivi:
L'algoritmo predefinito è `caching_sha2_password`.

Questo metodo utilizza l'hashing **SHA-256**. È considerato molto sicuro perché:

- Utilizza il "*salting*" (un valore casuale aggiunto alla password prima dell'hashing) per prevenire attacchi basati su tabelle precomputate (*Rainbow Tables*);
- Sfrutta la cache lato server per velocizzare le autenticazioni successive.

Password e sicurezza (2)

MySQL 8.0 e successivi:
L'algoritmo predefinito è `caching_sha2_password`.

Questo metodo utilizza l'hashing **SHA-256**. È considerato molto sicuro perché:

- Utilizza il "*salting*" (un valore casuale aggiunto alla password prima dell'hashing) per prevenire attacchi basati su tabelle precomputate (*Rainbow Tables*);
- Sfrutta la cache lato server per velocizzare le autenticazioni successive.

Password e sicurezza (2)

MySQL 8.0 e successivi:
L'algoritmo predefinito è `caching_sha2_password`.

Questo metodo utilizza l'hashing **SHA-256**. È considerato molto sicuro perché:

- Utilizza il "*salting*" (un valore casuale aggiunto alla password prima dell'hashing) per prevenire attacchi basati su tabelle precomputate (*Rainbow Tables*);
- Sfrutta la cache lato server per velocizzare le autenticazioni successive.

Password e sicurezza (3)

MySQL 5.7 e precedenti:
L'algoritmo predefinito era `mysql_native_password`.

Questo metodo utilizza un doppio hashing **SHA-1**. Sebbene sia ancora supportato per motivi di compatibilità, è considerato meno sicuro rispetto agli standard attuali perché SHA-1 è vulnerabile ad attacchi di collisione e forza bruta più rapidi.

Password e sicurezza (4)

È possibile specificare l'algoritmo che si vuole usare per la cifratura della password all'atto della creazione dell'utente:

```
CREATE USER 'utente_esempio'@'localhost'
IDENTIFIED WITH mysql_native_password BY 'password_sicura';
```

Q: Un esempio di quando potrebbe servire l'uso di un algoritmo di cifratura della password diverso da quello predefinito?

Password e sicurezza (4)

È possibile specificare l'algoritmo che si vuole usare per la cifratura della password all'atto della creazione dell'utente:

```
CREATE USER 'utente_esempio'@'localhost'
IDENTIFIED WITH mysql_native_password BY 'password_sicura';
```

Q: Un esempio di quando potrebbe servire l'uso di un algoritmo di cifratura della password diverso da quello predefinito?

Esercizio #1

Si scrivano le query SQL per creare i seguenti utenti:

1. Utente `mario.rossi` con password `mario.rossi123` che si può collegare solo dal server;
2. Utente `luigi.bianchi` con password `luigi.bianchi123` che si può collegare da qualsiasi host;
3. Utente `giuseppe.verdi` con password `giuseppe.verdi123` che si può collegare dalla sottorete `192.168.0.0/24`.

Tempo a disposizione: 10 minuti.

Indice

1. Introduzione
   - Creazione di un utente
   - Password e sicurezza
   - Esercizio #1
2. Gestione dei privilegi
   - I privilegi in MySQL
   - Privilegi sui dati (Data privileges)
3. Privilegi amministrativi
   - Esercizio #2
4. Esercizio #3

I privilegi in MySQL (1)

In MySQL ciò che un utente può o non può fare viene definito da un'insieme di permessi chiamati "*privilegi*".

I privilegi in MySQL vengono raggruppati per tipologia:

1. Privilegi `Data`:
   `SELECT`, `INSERT`, `UPDATE`, `DELETE` e `FILE`;
2. Privilegi `Structure`:
   `CREATE`, `ALTER`, `INDEX`, `DROP`, `CREATE TEMPORARY TABLES`, `SHOW VIEW`, `CREATE ROUTINE`, `ALTER ROUTINE`, `EXECUTE`, `CREATE VIEW`, `EVENT` e `TRIGGER`;

I privilegi in MySQL (1)

In MySQL ciò che un utente può o non può fare viene definito da un'insieme di permessi chiamati "*privilegi*".

I privilegi in MySQL vengono raggruppati per tipologia:

1. Privilegi `Data`:
   `SELECT`, `INSERT`, `UPDATE`, `DELETE` e `FILE`;
2. Privilegi `Structure`:
   `CREATE`, `ALTER`, `INDEX`, `DROP`, `CREATE TEMPORARY TABLES`, `SHOW VIEW`, `CREATE ROUTINE`, `ALTER ROUTINE`, `EXECUTE`, `CREATE VIEW`, `EVENT` e `TRIGGER`;

I privilegi in MySQL (1)

In MySQL ciò che un utente può o non può fare viene definito da un'insieme di permessi chiamati "*privilegi*".

I privilegi in MySQL vengono raggruppati per tipologia:

1. Privilegi `Data`:
   `SELECT`, `INSERT`, `UPDATE`, `DELETE` e `FILE`;
2. Privilegi `Structure`:
   `CREATE`, `ALTER`, `INDEX`, `DROP`, `CREATE TEMPORARY TABLES`, `SHOW VIEW`, `CREATE ROUTINE`, `ALTER ROUTINE`, `EXECUTE`, `CREATE VIEW`, `EVENT` e `TRIGGER`;

I privilegi in MySQL (2)

3. Privilegi `Administration`:
   `GRANT`, `SUPER`, `PROCESS`, `RELOAD`, `SHUTDOWN`, `SHOW DATABASES`, `LOCK TABLES`, `REFERENCES`, `REPLICATION CLIENT`, `REPLICATION SLAVE` e `CREATE USER`.

Tratto da: [1]

I privilegi in MySQL (3)

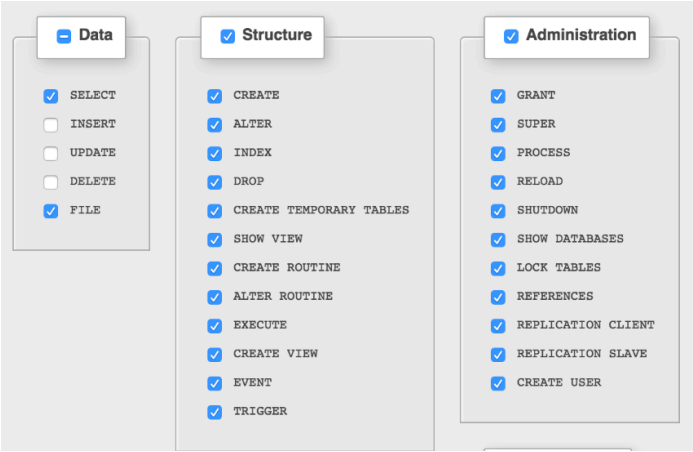

Figura 2.1: Gestione dei permessi in phpMyAdmin

Privilegi sui dati (Data privileges) (1)

Questi permessi regolano l'accesso e la modifica dei record all'interno delle tabelle.

- `SELECT` **(Lettura):**
  - **Cosa fa:** Permette all'utente di leggere i dati. È il permesso che autorizza l'esecuzione delle query standard per visualizzare i record;
  - **Utilizzo tipico:** È il privilegio base. Quasi ogni utente di un'applicazione ha bisogno di questo permesso per poter "vedere" i dati (ad esempio, per caricare i prodotti in un e-commerce o visualizzare il proprio profilo).

Privilegi sui dati (Data privileges) (2)

- `INSERT` **(Inserimento):**
  - **Cosa fa:** Consente di aggiungere nuove righe (record) all'interno di una tabella esistente;
  - **Utilizzo tipico:** Permette azioni come la registrazione di un nuovo utente, il salvataggio di un nuovo ordine o l'aggiunta di un commento in un blog.
- `UPDATE` **(Modifica):**
  - **Cosa fa:** Autorizza la modifica dei dati già presenti in una o più righe di una tabella;
  - **Utilizzo tipico:** Usato quando un utente cambia la propria password, aggiorna il proprio indirizzo o quando un amministratore modifica il prezzo di un prodotto.

Q: In quale sotto-insieme di SQL siamo?

Privilegi sui dati (Data privileges) (2)

- `INSERT` **(Inserimento):**
  - **Cosa fa:** Consente di aggiungere nuove righe (record) all'interno di una tabella esistente;
  - **Utilizzo tipico:** Permette azioni come la registrazione di un nuovo utente, il salvataggio di un nuovo ordine o l'aggiunta di un commento in un blog.
- `UPDATE` **(Modifica):**
  - **Cosa fa:** Autorizza la modifica dei dati già presenti in una o più righe di una tabella;
  - **Utilizzo tipico:** Usato quando un utente cambia la propria password, aggiorna il proprio indirizzo o quando un amministratore modifica il prezzo di un prodotto.

Q: In quale sotto-insieme di SQL siamo?

Privilegi sui dati (Data privileges) (2)

- `INSERT` **(Inserimento):**
  - **Cosa fa:** Consente di aggiungere nuove righe (record) all'interno di una tabella esistente;
  - **Utilizzo tipico:** Permette azioni come la registrazione di un nuovo utente, il salvataggio di un nuovo ordine o l'aggiunta di un commento in un blog.
- `UPDATE` **(Modifica):**
  - **Cosa fa:** Autorizza la modifica dei dati già presenti in una o più righe di una tabella;
  - **Utilizzo tipico:** Usato quando un utente cambia la propria password, aggiorna il proprio indirizzo o quando un amministratore modifica il prezzo di un prodotto.

Q: In quale sotto-insieme di SQL siamo?

Privilegi sui dati (Data privileges) (3)

- `DELETE` **(Cancellazione):**
  - **Cosa fa:** Permette di rimuovere intere righe da una tabella;
  - **Utilizzo tipico:** Serve per eliminare definitivamente un record, come cancellare un messaggio o rimuovere un prodotto dal catalogo.

Domanda "stile esame"

Una volta eseguite la `DELETE` il/i record viene/vengono permanentemente eliminato/i. Immagina di essere una PA (Pubblica Amministrazione) dove la possibilità di recuperare i dati accidentalmente cancellati è una necessità importante, come si può risolvere?

Privilegi sui dati (Data privileges) (3)

- `DELETE` **(Cancellazione):**
  - **Cosa fa:** Permette di rimuovere intere righe da una tabella;
  - **Utilizzo tipico:** Serve per eliminare definitivamente un record, come cancellare un messaggio o rimuovere un prodotto dal catalogo.

Domanda "stile esame"

Una volta eseguite la `DELETE` il/i record viene/vengono permanentemente eliminato/i. Immagina di essere una PA (Pubblica Amministrazione) dove la possibilità di recuperare i dati accidentalmente cancellati è una necessità importante, come si può risolvere?

Indice

1. Introduzione
   - Creazione di un utente
   - Password e sicurezza
   - Esercizio #1
2. Gestione dei privilegi
   - I privilegi in MySQL
   - Privilegi sui dati (Data privileges)
3. Privilegi amministrativi
   - Esercizio #2
4. Esercizio #3

Privilegi amministrativi (1)

Come dice il nome, i privilegi amministrativi sono quei privilegi che riguardano l'amministrazione del server in generale e non di una singola base di dati.

Uno dei più importanti è sicuramente il `GRANT`**:** esso permette di dare dei permessi a un altro utente su una base di dati / tabelle.

Per aggiungere il privilegio di `SELECT` vale:

```
GRANT SELECT ON db.* TO utente@localhost;
```

Q: *cosa notiamo dalla query?*

Privilegi amministrativi (1)

Come dice il nome, i privilegi amministrativi sono quei privilegi che riguardano l'amministrazione del server in generale e non di una singola base di dati.

Uno dei più importanti è sicuramente il `GRANT`: esso permette di dare dei permessi a un altro utente su una base di dati / tabelle.

Per aggiungere il privilegio di `SELECT` vale:

```
GRANT SELECT ON db.* TO utente@localhost;
```

Q: *cosa notiamo dalla query?*

Privilegi amministrativi (2)

Analizziamo la struttura della query:

```
GRANT <privilegi> ON <database>.<tabella> TO <user>@<host>;
```

- `<privilegi>` indica l'insieme di privilegi che l'utente sta per ottenere, si può usare `ALL PRIVILEGES` [1] per garantire tutti i privilegi;
- `<database>` indica il database sul quale si vogliono garantire i privilegi, si può usare `*` per indicare tutti i database del server;
- `<tabella>` indica la tabella sulla quale si vogliono garantire i privilegi, si può usare `*` per indicarle tutte;

[1] Approfondimento: [2]

Privilegi amministrativi (3)

```
GRANT <privilegi> ON <database>.<tabella> TO <user>@<host>;
```

- `<user>` indica l'utente al quale stiamo assegnando i privilegi;
- `<host>` indica l'host dell'utente.

Attenzione!

Abbiamo visto che per ogni utente è possibile specificare l'host/subnet dal quale si può connettere. Bisogna però fare attenzione che MySQL tratta le coppie di utente/host come entità distinte!

Privilegi amministrativi (3)

```
GRANT <privilegi> ON <database>.<tabella> TO <user>@<host>;
```

- `<user>` indica l'utente al quale stiamo assegnando i privilegi;
- `<host>` indica l'host dell'utente.

Attenzione!

Abbiamo visto che per ogni utente è possibile specificare l'host/subnet dal quale si può connettere. Bisogna però fare attenzione che MySQL tratta le coppie di utente/host come entità distinte!

Privilegi amministrativi (3)

Ad esempio, i seguenti utenti:

- `mario@localhost`;
- `mario@10.0.0.0/8`.

Anche se con lo stesso nome utente (`user`) verranno trattati come due entità distinte da MySQL. In caso di omonimia, MySQL cerca la corrispondenza migliore.

Domanda "stile esame"

Considera i due utenti scritti sopra, che tipo di controllo sulla corrispondenza può essere fatto da MySQL?

Privilegi amministrativi (3)

Ad esempio, i seguenti utenti:

- `mario@localhost`;
- `mario@10.0.0.0/8`.

Anche se con lo stesso nome utente (`user`) verranno trattati come due entità distinte da MySQL. In caso di omonimia, MySQL cerca la corrispondenza migliore.

Domanda "stile esame"

Considera i due utenti scritti sopra, che tipo di controllo sulla corrispondenza può essere fatto da MySQL?

Esercizio #2

Si consideri l'**Esercizio #1** della lezione del 26/02 (link).

Creare due utenti sul server MySQL:

1. Utente `enel_lettura` con privilegi: `SELECT`;
2. Utente `enel_scrittura` con privilegi: `SELECT`, `INSERT`, `UPDATE` e `DELETE`.

Modificare il proprio codice e rispondere poi ai seguenti quesiti:

- Che tipo di errore viene generato da PHP?
- Il codice che hai scritto è *error-sensitive*[1]? Se no, come si può correggere?

Tempo a disposizione: 15 minuti.

[1] Gestisce gli errori senza un "crash"

Indice

1. Introduzione
   - Creazione di un utente
   - Password e sicurezza
   - Esercizio #1
2. Gestione dei privilegi
   - I privilegi in MySQL
   - Privilegi sui dati (Data privileges)
3. Privilegi amministrativi
   - Esercizio #2
4. Esercizio #3

Esercizio #3 (1)

Un'azienda *start-up* vuole costruire una piattaforma Web che consenta il *car pooling* tra viaggiatori sul territorio nazionale, con l'obiettivo di diffondere l'uso di una mobilità flessibile e personalizzata in termini di percorsi e costi.

Gli utenti della piattaforma possono essere di due tipi: autisti (coloro che offrono un passaggio con la propria macchina) e passeggeri (coloro che usufruiscono del passaggio).

Gli autisti devono registrarsi sul sito ed inserire i propri dati: generalità (nome e cognome), numero della patente di guida, dati dell'automobile utilizzata, recapito telefonico ed e-mail.

Esercizio #3 (2)

Per ogni viaggio che intendono condividere, gli autisti devono indicare città di partenza, città di destinazione, data ed ora di partenza, contributo economico richiesto ad ogni passeggero e tempi di percorrenza stimati.

È responsabilità dell'autista, mano a mano che accetterà passeggeri per un certo viaggio, dichiarare chiuse le prenotazioni per quel viaggio, utilizzando un'apposita funzione sul portale.

Il passeggero si deve registrare sulla piattaforma, indicando cognome e nome, documento di identità, recapito telefonico ed email. La piattaforma fornisce ai passeggeri la possibilità di indicare città di partenza e di destinazione e data desiderata; presenta quindi un elenco di viaggi (per cui non siano ancora chiuse le prenotazioni), ciascuno con le caratteristiche dell'autista e le modalità del viaggio stesso inserite dall'autista (orario, eventuali soste previste alle stazioni di servizio, possibilità di caricare bagaglio o animali, . . .).

Esercizio #3 (3)

Lo studente, fatte le opportune ipotesi aggiuntive, sviluppi:

- Un'analisi della realtà di riferimento, giungendo alla creazione della base di dati che, a suo motivato giudizio, sia idoneo a gestire la realtà presentata;
- Il progetto di massima della struttura funzionale dell'applicazione Web, realizzando, con appropriati linguaggi a scelta sia lato client che lato server, un segmento significativo dell'applicazione che consente l'interazione con la base di dati.

Tratto da: [3]

Riferimenti e approfondimenti

[1] MySQL developers, *Privileges Provided by MySQL*, [Online; accessed 4-March-2026], 2026. indirizzo: https://dev.mysql.com/doc/refman/8.4/en/privileges-provided.html.

[2] 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.

[3] Ministero dell'Istruzione, dell'Università e della Ricerca, *Esame di Istruzione Secondaria Superiore: Tema di INFORMATICA*, 2017.