DOCUMENTAZIONE

Surricchio Noemi 254811 - Farinelli Alessio 254061 - Marono Vincenzo 254457 -
Belli Simone 254437 - Taccone Shandor 252625

INDICE

SPECIFICA DEI REQUISITI

CONTESTUALIZZAZIONE DELLA SPECIFICA

È stato deciso di realizzare un catalogo bibliografico online il cui utilizzo è limitato agli studenti dell'università dell'aquila. Il sistema è pensato per fornire agli studenti informazioni riguardanti sia i libri di testo sia libri che possano aiutare lo studio di una determinata materia anche se non consigliati dal professore del corso, pertanto sarà possibile loro aggiungere, recensire e commentare le pubblicazioni al fine di fornire informazioni utili e aprire dibattiti costruttivi con gli altri studenti.

INTEGRAZIONE DELLA SPECIFICA

La Biblioteca rappresenta un semplice gestore di catalogo bibliografico online destinato agli studenti dell'università dell'Aquila. Il catalogo conterrà una serie di pubblicazioni, ciascuna delle quali caratterizzata da un titolo, una lista di autori, un editore, una serie di metadati (vedi dopo) e, opzionalmente, una descrizione testuale (sunto del contenuto o presentazione della pubblicazione), un indice (composto dai titoli dei vari capitoli/sezioni della pubblicazione, numerati e ordinati) e una serie di sorgenti tramite le quali poter accedere alla pubblicazione o a parti di essa. Ciascuna sorgente sarà caratterizzata da un tipo, una URI, da un formato e da una descrizione. Esempi di sorgenti valide potrebbero essere i seguenti:

- *tipo="immagine", URI="http://server.net/cover.jpg", formato="image/jpeg", descrizione="copertina"*

- *tipo="download", URI="http://server.net/book.pdf", formato="application/pdf", descrizione="versione elettronica gratuita"*

- *tipo="acquisto", URI="http://www.amazon.it/xyz", formato="cartaceo, copertina rigida", descrizione="acquista online"*

Per quanto riguarda invece i metadati, questi saranno costituiti (almeno) dalle seguenti informazioni: codice ISBN, numero di pagine, materia (che se non sarà presente all'aggiunta di una pubblicazione potrà essere creata dall'utente stesso e poi accettata dal moderatore insieme alla pubblicazione), lingua, data di pubblicazione, lista delle ristampe (numero e data) e parole chiave identificative (zero o più). Inoltre, saranno indicate anche eventuali riedizioni con la data ed i cambiamenti apportati. Ciascuna pubblicazione potrà essere associata a un certo numero di recensioni testuali e like ed entrambi dovranno indicare l'utente che li ha inseriti, la data e l'ora dell'inserimento. A loro volta, le recensioni, potranno essere commentate dagli utenti e ogni commento potrà avere una serie di risposte. Sia recensioni sia pubblicazioni avranno un indice di gradimento espresso dagli utenti. Le recensioni saranno moderate, quindi è necessario prevedere un flag che indichi l'approvazione di essa da parte dei moderatori. Non è prevista un'approvazione per i commenti e le risposte ma sarà possibile mandare delle segnalazioni da parte degli utenti attivi per eventuali commenti e risposte non opportune. Nel momento della segnalazione di uno di questi, qualora il moderatore lo ritenesse necessario, l' utente potrà essere bannato per un periodo di 30 giorni (soft ban). Al raggiungimento di 3 soft ban l' utente verrà bannato dal sistema in maniera permanente (permaban), mentre alla totalità di 3 segnalazioni infondate verrà bannato (soft ban). Infine, gli utenti potranno anche modificare presentazioni, pubblicazioni, recensioni, commenti e risposte (presentazioni, pubblicazioni e recensioni richiederanno l' approvazione da parte di un moderatore anche in caso di modifica). Sarà inoltre necessario prevedere opportune strutture per immagazzinare i dati dell'utenza. Il sistema, infatti, prevede due tipologie di utenza: attiva e passiva. Gli utenti potranno registrarsi nel sito fornendo i loro dati anagrafici, un indirizzo email (che verrà usato anche come username) e, ovviamente, la password scelta. Il livello di utenza iniziale sarà sempre quello passivo. Ogni utente potrà scegliere di inserire una descrizione/presentazione personale riguardante l'attività di lettore che determinerà il passaggio da utente passivo a utente attivo qualora venisse approvata da un moderatore. L'utente moderatore sarà avvisato tramite una notifica quando verranno segnalati commenti e risposte, o per l'approvazione di una presentazione, recensione o pubblicazione e, se lo riterrà opportuno, potrà modificare il testo di un commento/risposta e questo

prenderà il suo nome al fine di non interrompere l'integrità dei commenti sottostanti. Per tener traccia delle modifiche effettuate alla bibliografia dai vari utenti attivi, il sistema dovrà mantenere una "storia" di ciascuna scheda bibliografica. Ogni entry di tale storia conterrà un timestamp, il nome dell'utente e una descrizione della modifica. La storia disporrà di filtri per ottenere le informazioni relative ad una specifica pubblicazione (aggiunta, modifica, eliminazione, like e recensione). Potete scegliere voi il dettaglio di questa descrizione: di base, sono accettabili anche descrizioni del tipo "ha modificato la pubblicazione", "ha inserito la pubblicazione", "ha approvato una recensione dell'utente X", ecc. Il sistema conterrà anche una "graduatoria top commenti" che si aggiornerà ogni settimana e che si baserà sulla somma tra il rapporto del numero di stelle e il numero di recensioni, con il rapporto del numero di recensioni ed una costante.Ci sono indubbiamente molti vincoli che possono essere applicati ai contenuti di questa base di dati. Ad esempio, non dovrebbe essere possibile per un utente inserire più di una recensione per la stessa pubblicazione. L'individuazione dei vincoli e la loro implementazione (con vincoli sulle tabelle, trigger o quantomeno definendo il codice e le query necessari a effettuarne il controllo) costituiscono un requisito importante per lo sviluppo di un progetto realistico, e ne verrà tenuto conto durante la valutazione finale.

INDIVIDUAZIONE DEI CONCETTI FONDAMENTALI

La Biblioteca rappresenta un semplice gestore di catalogo bibliografico online destinato agli studenti dell'università dell'Aquila. Il catalogo conterrà una serie di pubblicazioni, ciascuna delle quali caratterizzata da un titolo, una lista di autori, un editore, una serie di metadati (vedi dopo) e, opzionalmente, una descrizione testuale (sunto del contenuto o presentazione della pubblicazione), un indice (composto dai titoli dei vari capitoli/sezioni della pubblicazione, numerati e ordinati) e una serie di sorgenti tramite le quali poter accedere alla pubblicazione o a parti di essa. Ciascuna sorgente sarà caratterizzata da un tipo, una URI, da un formato e da una descrizione. Esempi di sorgenti valide potrebbero essere i seguenti:

- *tipo="immagine", URI="http://server.net/cover.jpg", formato="image/jpeg", descrizione="copertina"*

- *tipo="download", URI="http://server.net/book.pdf", formato="application/pdf", descrizione="versione elettronica gratuita"*

- *tipo="acquisto", URI="http://www.amazon.it/xyz", formato="cartaceo, copertina rigida", descrizione="acquista online"*

Per quanto riguarda invece i metadati, questi saranno costituiti (almeno) dalle seguenti informazioni: codice ISBN, numero di pagine, materia (che se non sarà presente all'aggiunta di una pubblicazione potrà essere creata dall'utente stesso e poi accettata dal moderatore insieme alla pubblicazione), lingua, data di pubblicazione, lista delle ristampe (numero e data) e parole chiave identificative (zero o più). Inoltre, saranno indicate anche eventuali riedizioni con la data ed i cambiamenti apportati. Ciascuna pubblicazione potrà essere associata a un certo numero di recensioni testuali e like ed entrambi dovranno indicare l'utente che le ha inserite, la data e l'ora dell'inserimento. A loro volta, le recensioni, potranno essere commentate dagli utenti e ogni commento potrà avere una serie di risposte. Sia le recensioni sia le pubblicazioni avranno un indice di gradimento espresso dagli utenti. Le recensioni saranno moderate, quindi è necessario prevedere un flag che indichi l'approvazione di essa da parte dei moderatori. Non è prevista un'approvazione per i commenti e le risposte ma sarà possibile mandare delle segnalazioni da parte degli utenti attivi per eventuali commenti e risposte non opportune. Al momento della segnalazione di uno di questi, qualora il moderatore lo ritenesse necessario, l'utente potrà essere bannato per un periodo di 30 giorni (soft ban) e al raggiungimento di 3 di questi esso verrà permabannato (ban permanente). Un utente, dopo aver effettuato 3 segnalazioni immotivate, potrà ricevere da parte del moderatore un soft ban, anch'esso di 30 giorni. Infine sarà possibile modificare pubblicazioni, recensioni commenti e risposte (presentazioni, pubblicazioni e recensioni richiederanno l'approvazione da parte di un moderatore anche in caso di modifica) da parte dell'utente che le ha inserite. Sarà inoltre necessario prevedere opportune strutture per immagazzinare i dati dell'utenza. Il sistema, infatti, prevede due tipologie di utenza: attiva e passiva. Gli utenti potranno registrarsi nel sito fornendo i loro dati anagrafici, un indirizzo email (che verrà usato anche come username) e, ovviamente, la password scelta. Il livello di utenza iniziale sarà sempre quello passivo. Ogni utente potrà scegliere di inserire una descrizione/presentazione personale riguardante

l'attività di lettore che determinerà il passaggio da utente passivo a utente attivo qualora venisse approvata da un moderatore. L'utente moderatore sarà avvisato tramite una notifica quando verranno segnalati commenti e risposte, o per l'approvazione di una presentazione, recensione o pubblicazione e, se lo riterrà opportuno, potrà modificare il testo di un commento/risposta e questo prenderà il suo nome al fine di non interrompere l'integrità dei commenti sottostanti. Per tener traccia delle modifiche effettuate alla bibliografia dai vari utenti attivi, il sistema dovrà mantenere una "storia" di ciascuna scheda bibliografica. Ogni entry di tale storia conterrà un timestamp, il nome dell'utente e una descrizione della modifica. La storia disporrà di filtri per ottenere le informazioni relative ad una specifica pubblicazione (aggiunta, modifica, eliminazione, like e recensione). Potete scegliere voi il dettaglio di questa descrizione: di base, sono accettabili anche descrizioni del tipo "ha modificato la pubblicazione", "ha inserito la pubblicazione", "ha approvato una recensione dell'utente X", ecc. Il sistema conterrà anche una "graduatoria top commenti" che si aggiornerà ogni settimana e che si baserà sulla somma tra il rapporto del numero di stelle e il numero di recensioni, con il rapporto del numero di recensioni ed una costante.Ci sono indubbiamente molti vincoli che possono essere applicati ai contenuti di questa base di dati. Ad esempio, non dovrebbe essere possibile per un utente inserire più di una recensione per la stessa pubblicazione. L'individuazione dei vincoli e la loro implementazione (con vincoli sulle tabelle, trigger o quantomeno definendo il codice e le query necessari a effettuarne il controllo) costituiscono un requisito importante per lo sviluppo di un progetto realistico, e ne verrà tenuto conto durante la valutazione finale.

GLOSSARIO DEI TERMINI

TERMINI	DESCRIZIONE	SINONIMI	COLLEGAMENTI
BIBLIOTECA	Servizio finalizzato a soddisfare bisogni informativi.	Catalogo bibliografico.	.
PUBBLICAZIONE	Libro		Metadati, Sorgenti, Recensione, Utente, Like, Autore

TERMINI	DESCRIZIONE	SINONIMI	COLLEGAMENTI
AUTORE	Soggetto che si è occupato della stesura della pubblicazione.		Pubblicazione.
METADATI	Serie di concetti che caratterizzano una pubblicazione.		Pubblicazione.
SORGENTI	Sequenza di informazioni o collegamenti per poter accedere ad una certa pubblicazione o ad una parte di essa.		Pubblicazione.
RECENSIONI TESTUALI	Breve descrizione rilasciata da un utente attivo riguardo una certa pubblicazione.	Recensioni.	Utente, Pubblicazione, commento
UTENTE	Persona che si interfaccia direttamente con la base di dati.	Moderatore, utente attivo, utente passivo, Studente	Pubblicazione, Recensione, Like, commento, risposta
LIKE	Gradimento lasciato da un utente riguardo una certa pubblicazione.		Pubblicazione, Utente
STORIA	Azioni che un utente compie nel sistema (Commentare, recensire, mettere like ecc..).	Storico.	Pubblicazione
RIEDIZIONE	Aggiornamento di una pubblicazione.		Pubblicazione.
MATERIA	È la materia di una pubblicaizione		Pubblicazione

TERMINI	DESCRIZIONE	SINONIMI	COLLEGAMENTI
NOTIFICA	Un messaggio che arriva all'utente moderatore in cui viene specificato cosa dovrà fare.		Utente, Pubblicazione, Recensione, Commento, Risposta, Presentazione.
COMMENTO	Commento che un utente lascia ad una recensione		Utente, Recensione, Risposta
RISPOSTA	Risposta che un utente fa ad un commento.		Commento, Utente

FRASI DESCRITTIVE

Pubblicazione :

- *Ogni pubblicazione è caratterizzata da un titolo, una lista di autori, un editore, una serie di metadati (vedi dopo) e, opzionalmente, una descrizione testuale (sunto del contenuto o presentazione della pubblicazione), un indice (composto dai titoli dei vari capitoli/sezioni della pubblicazione, numerati e ordinati) e una serie di sorgenti tramite le quali poter accedere alla pubblicazione o a parti di essa.*

- *Ciascuna pubblicazione potrà essere associata a un certo numero di recensioni testuali e like.*

- *Le pubblicazioni avranno un indice di gradimento espresso dagli utenti.*

- *Sarà possibile modificare una pubblicazione da parte dell'utente che l'ha inserita.*

Autore :

- *Ogni pubblicazione è caratterizzata da un titolo, una lista di autori, un editore, una serie di metadati (vedi dopo) e, opzionalmente, una descrizione testuale (sunto del contenuto o presentazione della pubblicazione), un indice (composto dai titoli dei vari capitoli/sezioni della pubblicazione, numerati e ordinati) e una serie di sorgenti tramite le quali poter accedere alla pubblicazione o a parti di essa.*

Metadati :

- *I Metadati saranno costituiti (almeno) dalle seguenti informazioni: codice ISBN, numero di pagine, materia (che se non sarà presente all'aggiunta di una pubblicazione potrà essere creata dall'utente stesso e poi accettata dal moderatore insieme alla pubblicazione), lingua, data di pubblicazione, lista delle ristampe (numero e data) e parole chiave identificative (zero o più).*

Sorgenti :

- *Le sorgenti permetteranno di accedere alla pubblicazione o a parti di essa.*
- *Ciascuna sorgente sarà caratterizzata da un tipo, una URI, da un formato e da una descrizione.*

Recensioni :

- *Ciascuna pubblicazione potrà essere associata a un certo numero di recensioni testuali*
- *Le Recensioni, potranno essere commentate dagli utenti.*
- *Le recensioni avranno un indice di gradimento espresso dagli utenti.*
- *Le Recensioni potranno essere modificate dagli utenti che le hanno pubblicate.*
- *Non dovrebbe essere possibile per un utente inserire più di una recensione per la stessa pubblicazione.*

Utente :

- *Gli utenti potranno registrarsi nel sito fornendo i loro dati anagrafici, un indirizzo email (che verrà usato anche come username) e, ovviamente, la password scelta.*
- *Le recensioni testuali dovranno indicare l'utente che le ha inserite..*
- *Gli utenti potranno assegnare i loro like alle pubblicazioni.*
- *Il sistema prevede due tipologie di utenza : Attiva e Passiva. Gli utenti potranno registrarsi nel sito fornendo i loro dati anagrafici, un indirizzo email (username) e una password. Il livello di utenza iniziale sarà sempre quello passivo. Le recensioni saranno moderate da utenti che avranno privilegi superiori a utenti base (attivi o passivi). Questi sono gli utenti Moderatori che, insieme agli utenti Gestori, hanno il compito di approvare o meno recensioni, presentazione di un utente e aggiunta di una pubblicazione.*

- *Gli utenti possono aggiungere delle pubblicazioni ed eventualmente creare una nuova materia se la pubblicazione che si sta inserendo non appartiene a nessuna di quelle presenti.*

- *Gli utenti potranno aggiungere al massimo una recensione testuale per ogni pubblicazione, accompagnato da un grado di apprezzamento.*

- *Le recensioni potranno essere commentate dagli utenti che potranno anche inserire delle risposte ai commenti.*

- *Gli utenti attivi potranno segnalare i commenti e le risposte per eventuali contenuti inappropriati.*

- *Gli utenti Moderatori potranno bannare per un periodo di 30 giorni gli utenti segnalati per i contenuti dei propri commenti/recensioni.*

- *Gli utenti Moderatori potranno bannare per un periodo di 30 giorni gli utenti che raggiungono un numero di 3 segnalazioni inappropriate.*

-

- *Gli utenti Moderatori potranno bannare permanentemente gli utenti che raggiungono il numero di 3 soft ban.*

- *Ogni utente potrà scegliere di inserire una descrizione/presentazione personale riguardante l'attività di lettore che determinerà il passaggio da utente passivo a utente attivo qualora venisse approvata da un moderatore.*

- *L'utente moderatore sarà avvisato tramite una notifica quando verranno segnalati commenti e risposte, o per l'approvazione di una presentazione, recensione o pubblicazione e, se lo riterrà opportuno, potrà modificare il testo di un commento/risposta e questo prenderà il suo nome al fine di non interrompere l'integrità dei commenti sottostanti*

Like :

- Ogni like sarà associato a utente, data e ora.

Storia :

- La storia terrà traccia delle modifiche effettuate alle pubblicazioni.

- Ogni Entry della "storia" conterrà un timestamp, la matricola dell'utente, una descrizione della notifica, il tipo di azione fatta e su che tipo di entità.

Riedizione :

- Per ogni Pubblicazione saranno indicate anche eventuali riedizioni con la data ed i cambiamenti apportati.

Materia :

- i metadati, questi saranno costituiti (almeno) dalle seguenti informazioni: codice ISBN, numero di pagine, materia (che se non sarà presente all'aggiunta di una pubblicazione potrà essere creata dall'utente stesso e poi accettata dal moderatore insieme alla pubblicazione), lingua, data di pubblicazione, lista delle ristampe (numero e data) e parole chiave identificative (zero o più).

Notifica :

- L'utente moderatore sarà avvisato tramite una Notifica quando verranno segnalati commenti e risposte, o per l'approvazione di una presentazione, recensione o pubblicazione

Commento :

- *Le recensioni potranno essere Commentate dagli utenti.*
- *Ogni Commento potrà avere una serie di risposte.*
- *I Commenti non opportuni potranno essere segnalati ed eventualmente modificati da un moderatore.*
- *I Commenti potranno essere modificati dall'utente che li ha inseriti.*

Risposta :

- *Ogni commento potrà avere una serie di Risposte.*
- *Le risposte non opportune potranno essere segnalate ed eventualmente modificate da un moderatore.*
- *Le risposte potranno essere modificate dagli utenti che le hanno inserite.*

GRADI DI UTENZA

	PASSIVO	ATTIVO	MODERATORE	GESTORE
Inserimento pubblicazione	✓	✓	✓	✓
Inserire recensione	✓	✓	✓	✓
Inserire Commento	✓	✓	✓	✓
Inserire risposta	✓	✓	✓	✓
Lasciare un like	✓	✓	✓	✓
Modificare	✗	✓	✓	✓
Eliminazione	✗	✓	✓	✓
Approvazioni	✗	✗	✓	✓
Modificare commenti segnalati	✗	✗	✓	✓
Apprezzare recensione	✗	✓	✓	✓

PROGETTAZIONE CONCETTUALE

SCELTA STRATEGIA DI PROGETTAZIONE

La strategia di progettazione scelta è la TOP-DOWN. Questa è una strategia che si basa sul concetto teorico di "divide et impera", infatti si parte dall'obbiettivo finale a cui si vuole e, man mano che si va avanti esso viene diviso in tanti piccoli obiettivi che, una volta risolti ed uniti tutti porteranno alla conclusione del progetto.

SCELTA DELLE ENTITÀ E DELLE RELAZIONI

Promozione del concetto di "Pubblicazione" ad entità:

La frase:
"Il catalogo conterrà una serie di pubblicazioni, ciascuna delle quali caratterizzata da un titolo, una lista di autori, un editore, una serie di metadati (vedi dopo) e, opzionalmente, una descrizione testuale (sunto del contenuto o presentazione della pubblicazione), un indice (composto dai titoli dei vari capitoli/sezioni della pubblicazione, numerati e ordinati) e una serie di sorgenti tramite le quali poter accedere alla pubblicazione o a parti di essa" identifica i seguenti attributi per l'entità **Pubblicazione**

1. *Titolo*
2. *Autori*
3. *Editore*
4. *Materia (attributo multi-valore)*
5. *Descrizione testuale*
6. *Indice*
7. *Stato pubblicazione*

Inoltre è possibile stabilire le relazioni legate all'entità Pubblicazione attraverso le seguenti frasi

- *" Il catalogo conterrà una serie di pubblicazioni (…) e una serie di sorgenti tramite le quali poter accedere alla pubblicazione o a parti di essa. Ciascuna sorgente sarà caratterizzata da un tipo, una URI, da un formato e da una descrizione…".*

L' entità Pubblicazione sarà collegata all'entità **Sorgente** *attraverso la relazione* **Disporre**.

-"Il catalogo conterrà una serie di pubblicazioni, ciascuna delle quali caratterizzata da un titolo, una lista di autori, un editore, una serie di metadati… ".

Questa frase definisce il collegamento con l'entità **Metadato** *attraverso la relazione* **Descrivere**.

- *" Ciascuna pubblicazione potrà essere associata a un certo numero di recensioni testuali, le quali dovranno anche indicare l'utente che le ha inserite e la data/ora dell'inserimento. "*
- *" La storia disporrà di filtri per ottenere le informazioni relative ad una specifica pubblicazione (aggiunta, modifica, eliminazione, like e recensione)".*

Le entità **Recensione**, **Utente Attivo** *e* **Utente** *saranno collegate a* **Pubblicazione** *tramite rispettivamente le relazioni di* **Recensire**, **Aggiungere** *e* **Like**.

- *" Per tener traccia delle modifiche effettuate alla bibliografia dai vari utenti attivi, il sistema dovrà mantenere una "storia" di ciascuna scheda bibliografica… "*

Anche l' entità **Storia** *a sua volta è collegata all' entità Pubblicazione attraverso la relazione* **Tracciare**.

Promozione del concetto di "Metadato" ad entità:

La scelta di promuovere il metadato come entità e non come attributo composto ha come fine quello di snellire la tabella Pubblicazione e di facilitare la gestione dei dati in essa contenuti.

"Per quanto riguarda invece i metadati, questi saranno costituiti (almeno) dalle seguenti informazioni: codice ISBN, numero di pagine, lingua, data di pubblicazione, lista delle ristampe (numero e data) e parole chiave identificative (zero o più)" identifica una serie di attributi per l'entità Metadato

1. *ISBN*
2. *Numero pagine*
3. *Lingua*
4. *Data di pubblicazione*
5. *Lista delle ristampe (in fase di ristrutturazione, sarà poi deciso se trattarla come tabella o in altro modo)*
6. *Parole chiave*

Promozione del concetto di "Sorgente" ad entità:

Analogamente le motivazioni che hanno portato alla scelta di trattare la sorgente come entità e non come attributo sono le stesse del metadato.

"Ciascuna sorgente sarà caratterizzata da un tipo, una URI, da un formato e da una descrizione" identifica una serie di attributi per l'entità Sorgente:

1. *Tipo*
2. *URI*
3. *Formato*
4. *Descrizione*

Promozione del concetto di "Storia" ad entità:

"Per tener traccia delle modifiche effettuate alla bibliografia dai vari utenti attivi, il sistema dovrà mantenere una "storia" di ciascuna scheda bibliografica. Ogni entry di tale storia conterrà un timestamp, il nome dell'utente e una descrizione della modifica".

La suddetta frase definisce una serie di attributi per l'entità Storia:

1. *Timestamp*
2. *Testo*
3. *Azione*
4. *Tipo*

Promozione del concetto di "Recensione" ad entità:

"Le recensioni saranno moderate, quindi è necessario prevedere un flag che indichi se la recensione è stata approvata oppure no dai moderatori".

Questa frase identifica una serie di attributi per l'entità Recensione:

1. *Stato recensione*
2. *Testo*

"…le quali (recensioni) dovranno anche indicare l'utente che le ha inserite e la data/ora dell'inserimento".

Questa frase definisce la relazione **Rilasciare** *verso l'entità* **Utente**.

"Sia recensioni sia pubblicazioni avranno un indice di gradimento espresso dagli utenti".

La suddetta frase definisce un ulteriore relazione verso l'entità **Utente** *ovvero la relazione* **Apprezzare**.

"Le recensioni, potranno essere commentate dagli utenti".
Questa frase definisce la relazione **Commentare** *verso l'entità* **Commento**.

Promozione del concetto di "Commento" ad entità:

L' entità Commento dispone di un solo attributo:

1. *Testo*

La frase:

"A loro volta, le recensioni, potranno essere commentate dagli utenti e ogni commento potrà avere una serie di risposte."

Definisce due collegamenti:
1. *Il collegamento con l'entità Utente attraverso la relazione* **Fare**
2. *Il collegamento con l'entità Risposta attraverso la relazione* **Rispondere**

Promozione del concetto di "Risposta" ad entità:

L'entità Risposta è anch'essa provvista di un solo attributo:

1. *Testo*

La frase:
"A loro volta, le recensioni, potranno essere commentate dagli utenti e ogni commento potrà avere una serie di risposte."

Definisce la relazione **Scrivere** *verso l'entità* **Utente**.

Promozione del concetto di "Utente" ad entità:

"Gli utenti potranno registrarsi nel sito fornendo i loro dati anagrafici, un indirizzo e-mail (che verrà usato anche come username) e ovviamente la password scelta".

L'entità Utente è radicalizzata dai seguenti attributi:

1. *Nome*
2. *Cognome*
3. *Matricola*
4. *Data iscrizione*
5. *Numero ban*
6. *Ultimo accesso*
7. *E-Mail*
8. *Password*
9. *Numero di telefono*
10. *Stato ban*
11. *Attività (attributo composto) :*
    - *Numero commenti*
    - *Numero stelle*
    - *Numero recensioni*
    - *Numero like*

La frase :
"Il sistema prevede due tipologie di utenza: attiva e passiva".
Definisce una specializzazione dell'entità Utente.

Dettagli della gerarchia Utente:

Le specializzazioni dell'entità utente saranno:

1. *Utente passivo*
2. *Utente attivo*

Utente passivo:

Descrive un utente che non ha pubblicato la sua presentazione.

Utente attivo:

Descrive un utente che ha una presentazione accettata da un moderatore.
Questa entità sarà collegata con Pubblicazione attraverso la relazione **Aggiungere**.
Inoltre essa avrà delle specializzazioni sotto di essa:

1. *Utente moderatore*
2. *Utente gestore*

L'Utente Moderatore è la prima specializzazione di Utente Attivo.

L'Utente Gestore è la specializzazione dell'Utente Moderatore.
Utente Moderatore:

La frase:

" Al raggiungimento di 3 soft ban l'utente verrà bannato dal sistema in maniera permanente (permaban)".

Definisce una relazione verso **Utente Bannato** *che sarà composta da tutti gli utenti bannati permanentemente.*

La frase:

" L'utente moderatore sarà avvisato tramite una notifica quando verranno segnalati commenti e risposte, o per l'approvazione di una presentazione, recensione o pubblicazione… ".

Definisce una relazione verso **Notifica**.

Promozione dell'entità utente bannato:

L'utilizzo dell'entità Utente Bannato permette la definizione delle motivazioni, tramite gli attributi, che porteranno l'utente ad essere successivamente ed eventualmente bannato in modo permanente (perma ban). I quali attributi sono:

1. *Motivazione*
2. *Matricola*

Questa entità sarà collegata con **Utente Bannato** *tramite la relazione* **Eliminare**.

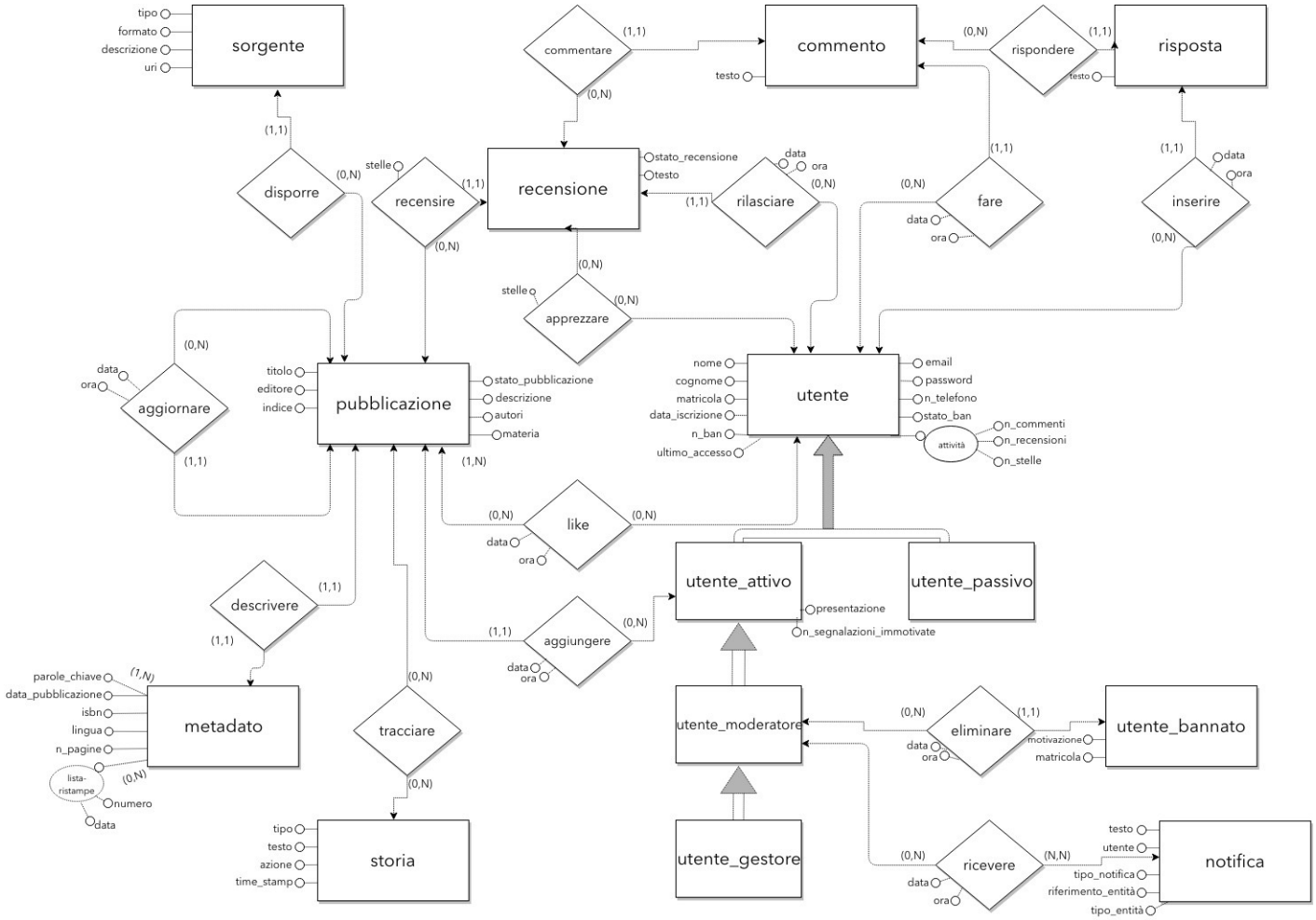
Promozione dell'entità notifica:

La frase:

" L'utente moderatore sarà avvisato tramite una notifica quando verranno segnalati commenti e risposte, o per l'approvazione di una presentazione, recensione o pubblicazione… ".

Descrive l'entità **Notifica** *che sarà collegata con l'entità Utente Moderatore tramite la relazione* **Ricevere**.

MODELLO E-R


sorgente
tipo
formato
descrizione
uri
disporre
(1,1)
(0,N)
commentare
(1,1)
(0,N)
commento
testo
rispondere
(0,N)
(1,1)
risposta
testo
recensire
stelle
(1,1)
(0,N)
recensione
stato_recensione
testo
rilasciare
data
ora
(1,1)
(0,N)
fare
(0,N)
(1,1)
data
ora
inserire
(1,1)
data
ora
(0,N)
apprezzare
stelle
(0,N)
(0,N)
aggiornare
data
ora
(0,N)
(1,1)
pubblicazione
titolo
editore
indice
stato_pubblicazione
descrizione
autori
materia
(1,N)
utente
nome
cognome
matricola
data_iscrizione
n_ban
ultimo_accesso
email
password
n_telefono
stato_ban
attività
n_commenti
n_recensioni
n_stelle
like
(0,N)
(0,N)
data
ora
descrivere
(1,1)
(1,1)
aggiungere
(1,1)
(0,N)
data
ora
utente_attivo
presentazione
n_segnalazioni_immotivate
utente_passivo
metadato
parole_chiave
(1,N)
data_pubblicazione
isbn
lingua
n_pagine
lista-ristampe
(0,N)
numero
data
tracciare
(0,N)
(0,N)
storia
tipo
testo
azione
time_stamp
utente_moderatore
utente_gestore
eliminare
(0,N)
(1,1)
data
ora
motivazione
matricola
utente_bannato
ricevere
(0,N)
(N,N)
data
ora
testo
utente
tipo_notifica
riferimento_entità
tipo_entità
notifica

DIZIONARIO DEI DATI - ENTITÀ

ENTITÀ	DESCRIZIONE	ATTRIBUTI
SORGENTE	Le sorgenti permettono di accedere alle pubblicazioni o a parti di esse	Tipo URI Formato Descrizione
METADATO	I metadati contengono eletti descrittivi per ogni pubblicazione	ISBN DataPubblicazione Lingua nPagine ParoleChiave ListaRistampe(Data, Numero)
RECENSIONE	La recensione è un testo valutativo e interpretativo di una pubblicazione che viene fatta da un utente.	Testo Stato_recensione
STORIA	La storia terrà traccia delle modifiche effettuate alla bibliografia degli utenti attivi. Ogni entry della storia conterrà un timestamp, il nome dell'utente e una descrizione della notifica	Testo Tipo Azione Time_stamp
PUBBLICAZIONE	Una pubblicazione può essere un opera letteraria, scientifica, artistica o di altro genere contenuta nella biblioteca.	Indice Titolo Autori Descrizione Materia Stato_pubblciazione Editore
UTENTE	Utente è la generalizzazione di "Gestore", "Utente Attivo", "Utente Passivo" e "Moderatore". È colui che usufruirà del servizio della biblioteca.	Nome Cognome n_elefono Email Password Matricola Data_iscrizione n_ban Ultimo_accesso Stato_ban Attività(n_commenti, n_recensioni, n_stelle)

ENTITÀ	DESCRIZIONE	ATTRIBUTI
UTENTE PASSIVO	È il grado di utenza iniziale.	
UTENTE ATTIVO	È il grado di utenza successivo a quello passivo.	Presentazione n_segnalazioni_immotivate
MODERATORE	É un utente attivo con permessi aggiuntivi	
GESTORE	Il Gestore è il grado più alto di utenza.	
UTENTE BANNATO	Utente bannato è un utente che ha ricevuto un numero di soft ban pari a 3. Egli non avrà più la possibilità di accedere al servizio in questione.	Motivazione Matricola
NOTIFICA	È una segnalazione che arriva al moderatore.	Testo Utente Tipo_notifica Riferimento_entità Tipo_entità
COMMENTO	Nota contestuale.	Testo
RISPOSTA	Avversione o approvazione di un commento.	Testo

DIZIONARIO DEI DATI - RELAZIONI

RELAZIONE	DESCRIZIONE	COMPONENTI	ATTRIBUTI
AGGIORNARE	Aggiornamento ovvero riedizione di una Pubblicazione.	Pubblicazione	Data Ora
DISPORRE	Una pubblicazione dispone di alcune Sorgenti.	Pubblicazione Sorgente	
RECENSIRE	Una pubblicazione viene recensita da una recensione.	Pubblicazione Recensione	Stelle

RELAZIONE	DESCRIZIONE	COMPONENTI	ATTRIBUTI
DESCRIVERE	Una pubblicazione è descritta da un metadato.	Metadato Pubblicazione	
RISPONDERE	Una risposta risponde ad un commento.	Risposta Commento	
LIKE	Un utente inserisce un like ad una pubblicazione.	Utente Pubblicazione	Data Ora
RILASCIARE	Un utente rilascia una. recensione.	Utente Recensione	Data Ora
APPREZZARE	Un utente apprezza una recensione.	Utente Recensione	Stelle
TRACCIARE	La storia traccia le pubblicazioni.	Pubblicazione Storia	
COMMENTARE	Una recensione è commentata da un commento.	Recensione Commento	
AGGIUNGERE	Utente attivo aggiunge una pubblicazione.	Utente Attivo Pubblicazione	Data Ora
FARE	Un utente fa un commento.	Utente Commento	Data Ora
INSERIRE	Un utente inserisce una risposta.	Utente Risposta	Data Ora
ELIMINARE	Un utente moderatore elimina un utente bannato.	Utente moderatore Utente bannato	Data Ora
RICEVERE	Un utente moderatore riceve delle notifiche.	Utente moderatore Notifica	Data Ora

VINCOLI NON ESPRIMIBILI DAL MODELLO E-R

Pubblicazione.

Vincolo di obbligatorietà.

Questo tipo di vincolo è presente nei seguenti attributi :

- *titolo*

```
    titolo varchar(100) NOT NULL,
```

- *editore*

```
    editore varchar(50) NOT NULL,
```

- *stato_pubblicazione*

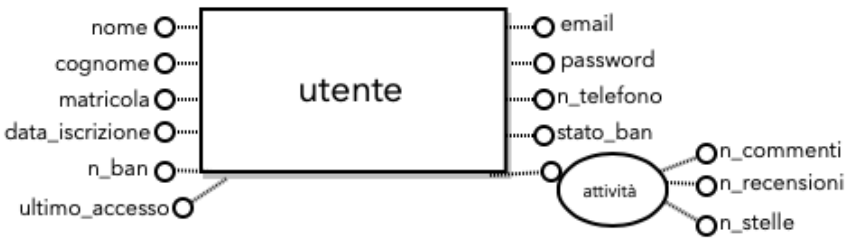
```
    stato_pubblicazione boolean NOT NULL
```

Altri Vincoli.

- *stato_pubblicazione deve essere di default 0*

```
stato_pubblicazione boolean NOT NULL default 0,    /* 0 = non accettata, 1 = accettata */
```

Utente.



Vincolo di obbligatorietà.

- *nome*

```
nome varchar(20) NOT NULL,
```

- *cognome*

```
cognome varchar(20) NOT NULL,
```

- *email*

```
email varchar(59) NOT NULL unique,
```

- *password*

```
password varchar(30) NOT NULL
```

- *n_ban*

```
n_ban int NOT NULL default 0,
```

- *ultimo_accesso*

```
ultimo_accesso date NOT NULL,
```

- *giorno_iscrizione*

```
giorno_iscrizione date NOT NULL,
```

- *n_commenti*

```
n_commenti int NOT NULL
```

- *n_recensioni*

```
n_recensioni int NOT NULL
```

- *n_stelle*

```
n_stelle int NOT NULL
```

Altri Vincoli.

- *l'email deve essere univoca*

```
email varchar(59) NOT NULL unique,
```

- *il n_telefono deve essere univoco*

```
n_telefono varchar(10) unique,
```

- *la password dovrà essere lunga almeno 8 caratteri*

```
password varchar(30) NOT NULL CONSTRAINT password_ck
CHECK(char_length(password) > 7),
```

- *il n_commenti deve essere maggiore o uguale a 0.*

```
n_commenti int NOT NULL default 0 CONSTRAINT n_commenti_ck
CHECK(n_commenti >= 0),
```

- *il n_recensioni deve essere maggiore o uguale a 0.*

```
n_recensioni int NOT NULL default 0 CONSTRAINT n_recensioni_ck
CHECK(n_recensioni >= 0),
```

- *il n_stelle deve essere maggiore o uguale a 0.*

```
n_stelle int NOT NULL default 0 CONSTRAINT n_stelle_ck
CHECK(n_stelle >= 0),
```

- *n_stelle deve avere 0 come valore di default*

```
n_stelle int NOT NULL default 0
```

- *n_recensioni deve avere 0 come valore di default*

```
n_recensioni int NOT NULL default 0
```

- *n_commenti deve avere 0 come valore di default*

```
n_commenti int NOT NULL default 0
```

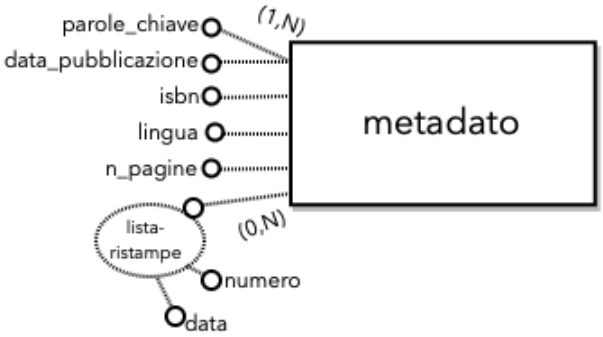
- *n_ban deve avere 0 come valore di default*

```
n_ban int NOT NULL default 0,
```

- *stato_ban deve avere null come valore di default*

```
stato_ban boolean default null,
```

Metadato.



Vincolo di obbligatorietà.

- *giorno_pubblicazione*

```
    giorno_pubblicazione date NOT NULL,
```

- *lingua*

```
    lingua enum('italiano', 'inglese', 'altro') NOT NULL,
```

- *n_pagine*

```
    n_pagine int NOT NULL
```

Altri Vincoli.

- *n_pagine deve essere maggiore stretto di 0*

```
n_pagine int NOT NULL CONSTRAINT n_pagine_ck CHECK(n_pagine > 0),
```

- *lingua potrà contenere solo uno dei seguenti valori: "italiano" "inglese" "altro"*

```
lingua enum('italiano', 'inglese', 'altro') NOT NULL,
```

Sorgente.

Vincolo di obbligatorietà.

- *uri*

```
uri varchar(2100) NOT NULL,
```

- *tipo*

```
tipo enum('immagine', 'download', 'acquisto', 'video', 'altro')
NOT NULL,
```

Altri Vincoli.

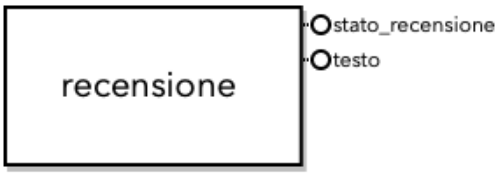
- *tipo dovrà contenere solo i seguenti valori: "immagine", "download", "acquisto", "video", "altro"*

```
tipo enum('immagine', 'download', 'acquisto', 'video', 'altro')
NOT NULL,
```

- *formato dovrà contenere solo i seguenti valori: "jpg", "pdf", "epub", "copertina rigida", "copertina flessibile", "youtube", "vimeo", "altro".*

```
formato enum('jpg', 'pdf', 'epub', 'copertina rigida', 'copertina
flessibile', 'youtube', 'vimeo', 'altro'),
```

Recensione.



Vincolo di obbligatorietà.

- *stato_recensione*

```
stato_recensione boolean NOT NULL
```

- *testo*

```
testo text NOT NULL,
```

Altri Vincoli.

- *stato_recensione deve essere di default 0*

```
stato_recensione boolean NOT NULL default 0,
```

Utente Attivo.

Vincolo di obbligatorietà.

- *n_segnalazioni_immotivate*

```
n_segnalazioni_immotivate int default 0 NOT NULL
```

- *presentazione*

```
presentazione text NOT NULL,
```

Altri Vincoli.

- *n_segnalazioni_immotivate può assumere solo un valore compreso tra 1 e 3*

```
n_segnalazioni_immotivate int default 0 NOT NULL CONSTRAINT
segnalazioni_min_0_max_3 CHECK(n_segnalazioni_immotivate <= 3 &&
n_segnalazioni_immotivate >= 0),
```

- *n_segnalazioni_immotivate deve essere di default zero*

```
n_segnalazioni_immotivate int default 0
```

Utente Bannato.

Vincolo di obbligatorietà.

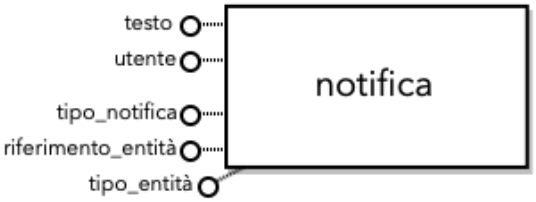
- *matricola*

```
matricola varchar(10) NOT NULL,
```

- *motivazione.*

```
motivazione text NOT NULL,
```

Notifica.



Vincolo di obbligatorietà.

- *testo*

```
testo text NOT NULL,
```

- *tipo_notifica*

```
tipo_notifica enum('segnalazione', 'approvazione') NOT NULL,
```

- *riferimento_entità*

```
riferimento_entita varchar(50) NOT NULL,
```

- *tipo_entità*

```
tipo_entita enum('commento', 'risposta', 'recensione',
'pubblicazione', 'presentazione') NOT NULL,
```

Altri Vincoli.

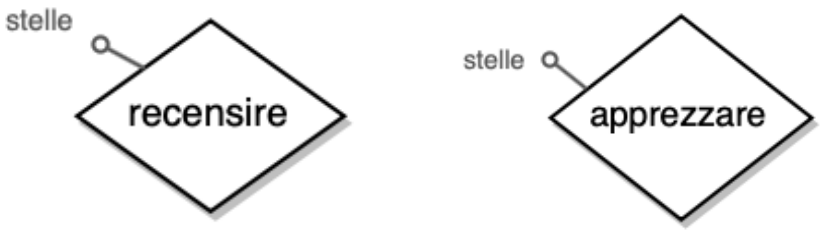
- *tipo_entita dovrà contenere solo i seguenti valori: "commento", "risposta", "recensione", "pubblicazione", "presentazione"*

```
tipo_entita enum('commento', 'risposta', 'recensione',
'pubblicazione', 'presentazione')
```

- *tipo_notifica dovrà contenere solo i seguenti valori: "segnalazione", "approvazione"*

```
tipo_notifica enum('segnalazione', 'approvazione')
```

Recensire e Apprezzare.



Vincolo di obbligatorietà.

- *stelle*

```
stelle int NOT NULL
```

Altri Vincoli.

- *stelle può assumere solo valore compreso tra 0 e 5*

```
stelle int NOT NULL CONSTRAINT stelle_ck2 CHECK(stelle >= 0 and
stelle < 6),
```

RISTRUTTURAZIONE

La fase di ristrutturazione di uno schema Entità-Relazionale si può suddividere in una serie di passi da effettuare in sequenza:

- *Analisi delle ridondanze*
- *Eliminazione delle generalizzazioni*
- *Partizionamento/Accorpamento delle entità*
- *Scelta degli identificatori principali*

ANALISI DELLE RIDONDANZE

Eliminazione ridondanze

Una ridondanza in uno schema concettuale corrisponde alla presenza di un attributo che può essere derivato da altri dati.

Durante questa fase prendiamo in considerazione la media del rapporto tra velocità di lettura e velocità di scrittura su disco delle memorie più comuni. Questo rapporto solitamente si aggira intorno al valore 2 perciò consideriamo che un operazione di lettura abbia complessità 1 mentre un operazione di scrittura abbia complessità 2.

Andiamo ora a vedere le ridondanze presenti nel nostro sistema.

Nell'Entità Utente troviamo le seguenti ridondanze :

n_commenti

L'attributo n_commenti rappresenta il numero di commenti che l'utente ha fatto e , nonostante nel nostro caso verrà mostrato soltanto nel profilo personale, è possibile che in implementazioni future possa essere utile per determinate operazioni, perciò è bene tenerne conto ugualmente. Questo valore può essere ricavato dalla relazione "fare" dove abbiamo proprio l'associazione tra l'utente e il commento lasciato da esso. Calcolare questo valore implicherebbe la seguente operazione :

$$(nu \cdot nc) \cdot cl = 150'000'000'000$$

dove :

- *nu è il numero di utenti = 50'000*
- *nc è il numero di commenti = 3'000'000*
- *cl è la complessità di lettura = 1*

La scelta di aggiungere questo attributo è stata fatta in quanto il calcolo precedente diventerebbe una semplice lettura del valore desiderato quindi avrebbe complessità pari a 1. Lo svantaggio di avere questo attributo potrebbe essere il costo di aggiornamento di quest'ultimo, che equivale a :

$$(ru \cdot nca \cdot cs) = 14'000$$

dove :

- *ru è il costo di ricerca di un utente data la sua matricola, questo costo è costante* $O(1)$ *in quanto la matricola è indicizzata.*
- *nca è il numero di commenti aggiunti settimanalmente = 7'000*
- *cs è il costo di scrittura = 2.*

n_recensioni e n_stelle

Questi due attributi rappresentano rispettivamente il numero di recensioni inserite da un utente e il numero di stelle ricevute per le proprie recensioni. Nel nostro sistema questi attributi sono utilizzati nell'operazione di calcolo della graduatoria top commenti in questo modo $(\frac{ns}{nr}) + (\frac{nr}{200})$ *dove ns e nr sono rispettivamente il numero di stelle e il numero di recensioni. La complessità del calcolo di questi due valori senza gli appositi attributi in utente sarebbe la seguente :*

numero di commenti per ogni utente

$$(nu \cdot nrec) \cdot cl = 1{,}25e11$$

numero di stelle per ogni utente

$$(nu \cdot nrec \cdot napp) \cdot cl = 3.125e17$$

dove :

- *nu è il numero di utenti = 50'000*
- *nrec è il numero di recensioni = 2'500'000*
- *napp è il numero di apprezzamenti = 2'500'000*
- *cl è la complessità di lettura = 1*

Avendo questi attributi nell'entità utente, le operazioni appena descritte non serviranno in quando potremo accedere a questi valori in tempo costante $O(1)$*. Lo svantaggio, come negli attributi descritti in precedenza è la complessità di aggiornamento delle variabili che riporteremo di seguito.*

aggiornamento della variabile n_recensioni

$$O(1)$$

aggiornamento della variabile n_stelle

$$O(1)$$

Aggiunta ridondanze

Nell'Entità Utente abbiamo aggiunto le seguenti ridondanze :

n_pubblicazioni

Questo attributo sta a rappresentare il numero di pubblicazioni aggiunte da un utente. Esaminando la query numero 4 (Estrazione dell'elenco degli utenti che hanno aggiunto più pubblicazioni) calcoliamo la complessità necessaria per l'output senza disporre di questo attributo.

$$nu \cdot np \cdot cl \cdot 4'000 = 1e13$$

dove :

- nu è il numero di utenti = 50'000
- np è il numero di pubblicazioni = 50'000
- cl è la complessità di lettura = 1
- 4'000 è la frequenza settimanale di chiamata alla query

La complessità di questa query con l'attributo n_pubblicazioni, invece, sarà

$$nu \cdot cl \cdot 4'000 = 4e8$$

dove :

- nu è il numero di utenti = 50'000
- cl è la complessità di lettura = 1
- 4'000 è la frequenza settimanale di chiamata alla query

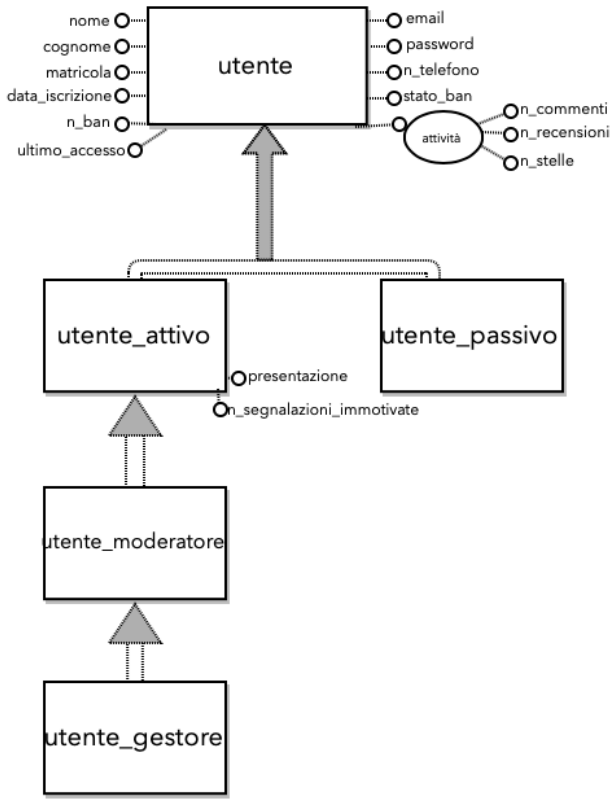
Il costo di aggiornamento dell'attributo, invece, sarà il seguente

$$O(1)$$

ELIMINAZIONE GENERALIZZAZIONI

Le generalizzazioni devono essere eliminate perché nei comuni DBMS relazionali non possono essere rappresentate.

L'unica generalizzazione presente nel nostro sistema è quella dell'entità **utente**.



*La modalità di eliminazione scelta è l'**accorpamento delle figlie nel genitore.** Le motivazioni di questa scelta sono le seguenti :*

- Presenza di molte relazioni che coinvolgono l'entità utente. Scegliendo di sostituire la generalizzazione con una relazione o di accorpare il genitore nelle figlie ci sarebbe stato l'effetto collaterale della duplicazione di queste relazioni.

- Presenza di pochi o nessun attributo aggiuntivo nelle entità figlie. Per questo non ha molto senso la loro esistenza come entità individuali, mentre basterebbero dei flag in utente per la distinzione.

- *Presenza di molti attributi nell'entità padre riguardanti tutte le figlie. Di conseguenza qualsiasi altra scelta avrebbe portato alla duplicazione di tutti gli attributi generando, anche a causa del punto precedente, delle entità individuali molto simili, perciò quasi prive di senso.*

ELIMINAZIONE DEGLI ATTRIBUTI MULTIVALORE

Gli attributi multivalore presenti nel nostro sistema sono i seguenti :

- ***attività*** *in Utente*
- ***lista_ristamp****e in Metadato. Questo attributo è stato trasformato nell'Entità Ristampe connessa con Pubblicazione tramite Venire.*

Questi attributi devono essere eliminati per lo stesso motivo delle generalizzazioni.

PARTIZIONAMENTO E ACCORPAMENTO DICONCETTI

Accorpamento dell'entità Metadato all'interno di Pubblicazione. Questa scelta è stata fatta in quanto, come nella query 8, potrebbe essere necessario accedere a tutte le informazioni relative ad una pubblicazione con una sola operazione, in questo modo eviteremo una JOIN fra le due entità coinvolte.

L'attributo composto parola_chiave è stato trasformato nell'Entità Parola_chiave connessa con Pubblicazione tramite Contenere.

L'attributo comporto materia è stato trasformato nell'Entità Materia connessa con Pubblicazione tramite Appartenere.

L'attributo comporto autore è stato trasformato nell'Entità Autore connessa con Pubblicazione tramite Scrivere.

SCELTA DEGLI IDENTIFICATORI PRINCIPALI

Le chiavi primarie delle relazioni sono le seguenti:

- *Utente: Matricola*
- *Pubblicazione: isbn*
- *Sorgente: id*
- *Materia: nome*
- *Ristampa: id*
- *Recensione: id*
- *Commento: id*
- *Risposta: id*
- *Notifica: id*
- *Utente_bannato: utente_matricola*
- *Storia: id*
- *Autore: id*
- *Parola_chiave: parola*

Non disponendo di attributi candidati a chiave nelle Entità Sorgente, Ristampa, Recensione, Commento, Risposta, Notifica, Storia e Autore sono stati utilizzati identificatori creati appositamente ("id").

TABELLA DEI VOLUMI

CONCETTO	TIPO	VOLUME
AGGIORNARE	R	30'000
APPREZZARE	R	2'500'000
BANNARE	R	5'000
COMMENTO	E	5'000'000
DESCRIVERE	R	50'000
DISPORRE	R	50'000
LIKE	R	5'000'000
METADATO	E	50'000
NOTIFICHE	E	4'000
PUBBLICAZIONE	E	50'000
RECENSIONE	E	2'500'000
RICEVERE	R	4'000
RILASCIARE	R	2'500'000
RISPONDERE	R	2'500'000
SORGENTE	E	250'000
STORIA	E	55'000
TRACCIARE	R	55'000
UTENTE	E	50'000
→GESTORE	E	10
→MODERATORE	E	100
→UTENTE ATTIVO	E	29'890
→UTENTE PASSIVO	E	20'000
UTENTE BANNATO	E	5'000

TABELLA DEI FORMATI

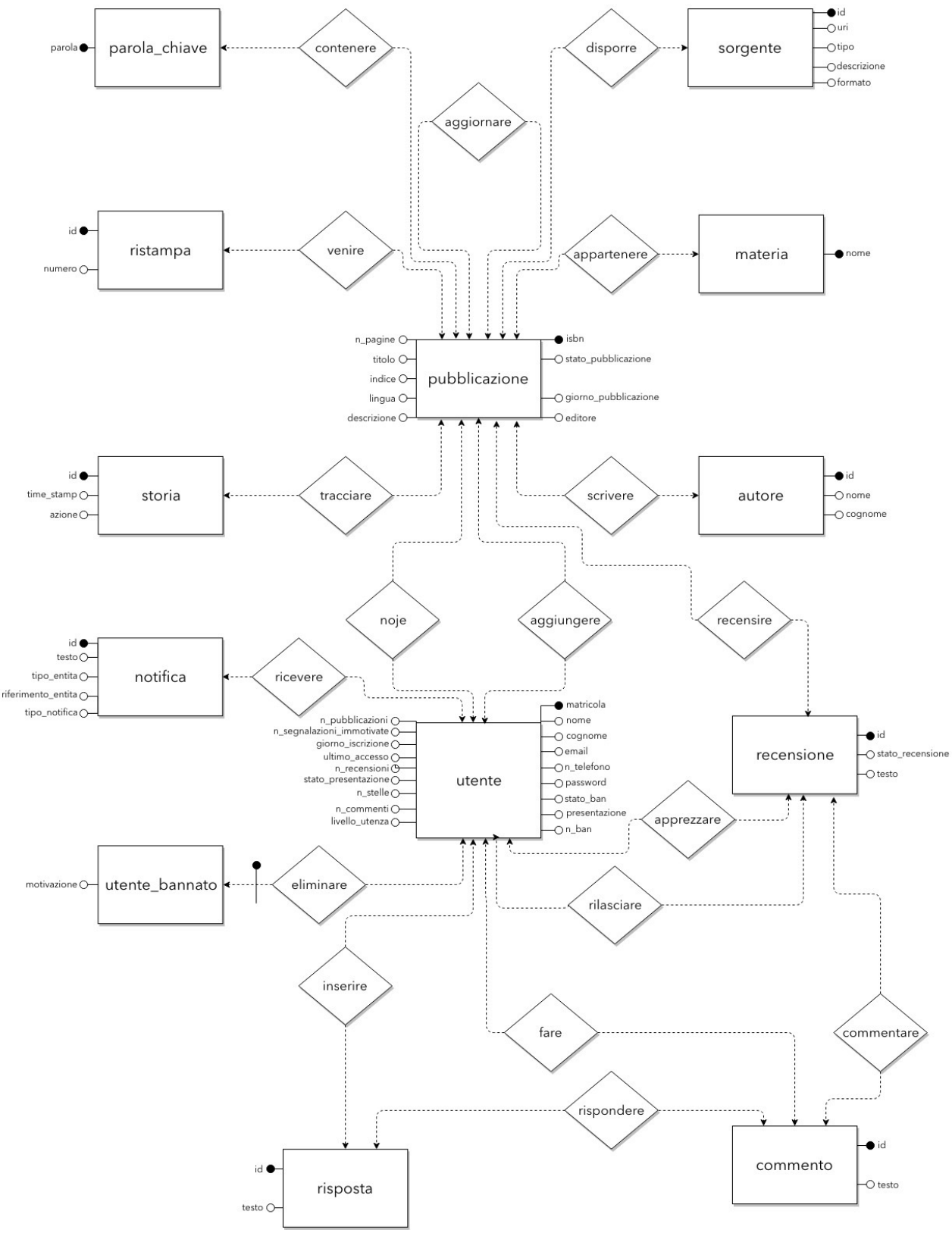
TABELLA	ATTRIBUTO	CHIAVE	FORMATO	DIMENSIONE
UTENTE	matricola	PK	carattere	10
	nome		carattere	20
	cognome		carattere	20
	email		carattere	59
	password		carattere	30
	n_telefono		carattere	10
	stato_ban		vero/falso	predefinito
	n_ban		intero	predefinito
	ultimo_accesso		data	predefinito
	giorno_iscrizione		data	predefinito
	n_commenti		intero	predefinito
	n_recensioni		intero	predefinito
	n_stelle		intero	predefinito
	n_pubblicazioni		intero	predefinito
	presentazione		testo	predefinito
	stato_presentazione		vero/falso	predefinito
	livello_utenza		vero/falso	predefinito
	n_segnalazioni_immotivate		intero	predefinito
PUBBLICAZIONE	isbn	PK	carattere	15
	giorno_pubblicazione		data	predefinito
	lingua		enum	3 valori
	n_pagine		intero	predefinito
	titolo		carattere	100
	descrizione		testo	predefinito
	indice		blob	predefinito
	editore		carattere	50

TABELLA	ATTRIBUTO	CHIAVE	FORMATO	DIMENSIONE
	stato_pubblicazione		vero/falso	predefinito
	utente_matricola	FK	carattere	10
SORGENTE	id	PK	intero	predefinito
	uri		carattere	2100
	tipo		enum	5 valori
	formato		enum	8 valori
	descrizione		testo	predefinito
	pubblicazione_isbn	FK	carattere	15
MATERIA	nome	PK	carattere	20
RISTAMPA	id	PK	intero	predefinito
	ora		ora	predefinito
	giorno		data	predefinito
	numero		intero	predefinito
	pubblicazione_isbn	FK	carattere	15
RECENSIONE	id	PK	intero	predefinito
	stato_recensione		vero/falso	predefinito
	testo		testo	predefinito
	stelle		intero	predefinito
	giorno		data	predefinito
	ora		ora	predefinito
	utente_matricola	FK	carattere	10
	pubblicazione_isbn	FK	carattere	15
COMMENTO	id	PK	intero	predefinito
	testo		testo	predefinito
	giorno		data	predefinito
	ora		ora	predefinito
	recensione_id	FK	intero	predefinito
	utente_matricola	FK	carattere	10
RISPOSTA	id	PK	intero	predefinito
	testo		testo	predefinito

TABELLA	ATTRIBUTO	CHIAVE	FORMATO	DIMENSIONE
	giorno		data	predefinito
	ora		ora	predefinito
	commento_id	FK	intero	predefinito
	utente_matricola	FK	carattere	
NOTIFICA	id	PK	intero	predefinito
	tipo_notifica		enum	2 valori
	tipo_entita		enum	5 valori
	riferimento_entita		carattere	50
	testo		testo	predefinito
UTENTE_BANNATO	utente_matricola	PK/FK	carattere	10
	motivazione		testo	predefinito
	giorno		data	predefinito
	ora		ora	predefinito
STORIA	id	PK	intero	predefinito
	azione		enum	3 valori
	time_stamp		timestamp	predefinito
	utente_matricola	FK	carattere	10
	pubblicazione_isbn	FK	carattere	15
AUTORE	id	PK	intero	predefinito
	nome		carattere	20
	cognome		carattere	20
PAROLA_CHIAVE	parola	PK	carattere	20
CONTENERE	parola_chiave	FK	carattere	20
	pubblicazione_isbn	FK	carattere	20
SCRIVERE	autore_id	FK	intero	predefinito
	pubblicazione_isbn	FK	carattere	15
AGGIORNARE	pubblicazione_isbn	FK	carattere	15
	riedizione_isbn	FK	carattere	15
APPARTENERE	materia_nome	FK	carattere	20
	pubblicazione_isbn	FK	carattere	15

TABELLA	ATTRIBUTO	CHIAVE	FORMATO	DIMENSIONE
APPREZZARE	utente_matricola	FK	carattere	10
	recensione_id	FK	intero	predefinito
NOJE	pubblicazione_isbn	FK	carattere	15
	utente_matricola	FK	carattere	10
	giorno		data	predefinito
	ora		ora	predefinito
RICEVERE	utente_matricola	FK	carattere	10
	notifica_id	FK	intero	predefinito
	giorno		data	predefinito
	ora		ora	predefinito

MODELLO E/R FINALE



PROGETTAZIONE LOGICA

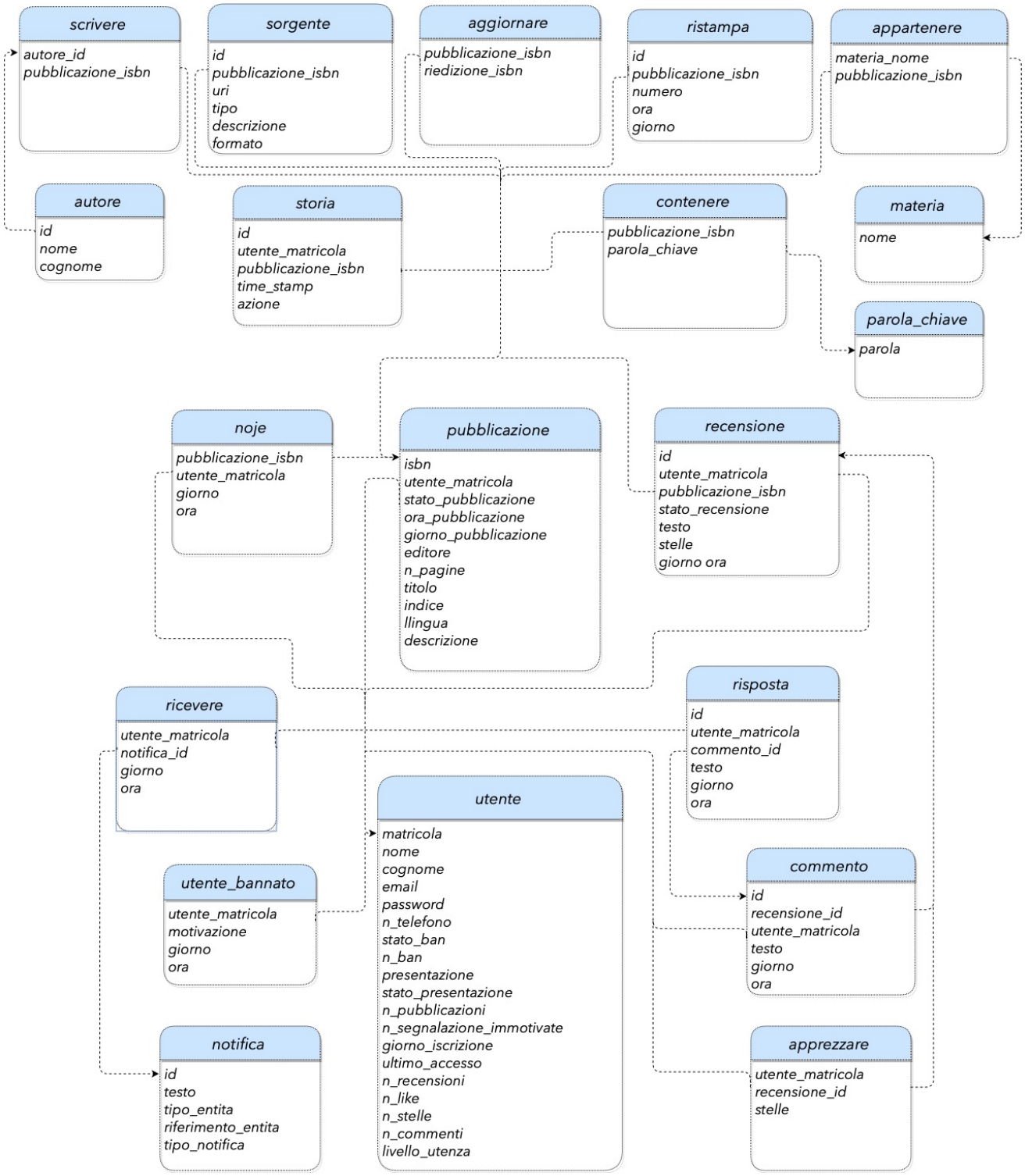
MODELLO LOGICO

Entità

- *utente (matricola, nome, cognome, email, password, n_telefono, stato_ban, n_ban, ultimo_accesso, giorno_iscrizione, n_commenti, n_recensioni, n_stelle, n_pubblicazioni, presentazione, stato_presentazione, livello_utenza, n_segnalazioni_immotivate)*
- *pubblicazione (isbn, giorno_pubblicazione, lingua, n_pagine, titolo, descrizione, indice, editore, stato_pubblicazione, utente_matricola)*
- *sorgente(id, uri, tipo, formato, descrizione, pubblicazione_isbn)*
- *materia (nome)*
- *ristampa (id, ora, giorno, numero, pubblicazione_isbn)*
- *recensione(id, stato_recensione, testo, stelle, giorno, ora, utente_matricola, pubblicazione_isbn)*
- *commento(id, testo, giorno, ora, recensione_id, utente_matricola)*
- *risposta (id, testo, giorno, ora, commento_id, utente_matricola)*
- *notifica (id, tipo_notifica, tipo_entita, riferimento_entita, testo)*
- *utente_bannato (utente_matricola, motivazione, giorno, ora)*
- *storia (id, azione, time_stamp, utente_matricola, pubblicazione_isbn)*
- *autore (id, nome, cognome)*
- *parola_chiave (parola)*

Relazioni

- *contenere (parola_chiave, pubblicazione_isbn)*
- *scrivere (autore_id, pubblicazione_isbn)*
- *aggiornare (pubblicazione_isbn, riedizione_isbn)*
- *appartenere (materia_nome, pubblicazione_isbn)*
- *apprezzare (utente_matricola, recensione_id, stelle)*
- *noje (pubblicazione_isbn, utente_matricola, giorno, ora)*
- *ricevere (utente_matricola, notifica_id, giorno, ora)*


scrivere
autore_id
pubblicazione_isbn
sorgente
id
pubblicazione_isbn
uri
tipo
descrizione
formato
aggiornare
pubblicazione_isbn
riedizione_isbn
ristampa
id
pubblicazione_isbn
numero
ora
giorno
appartenere
materia_nome
pubblicazione_isbn
autore
id
nome
cognome
storia
id
utente_matricola
pubblicazione_isbn
time_stamp
azione
contenere
pubblicazione_isbn
parola_chiave
materia
nome
parola_chiave
parola
noje
pubblicazione_isbn
utente_matricola
giorno
ora
pubblicazione
isbn
utente_matricola
stato_pubblicazione
ora_pubblicazione
giorno_pubblicazione
editore
n_pagine
titolo
indice
llingua
descrizione
recensione
id
utente_matricola
pubblicazione_isbn
stato_recensione
testo
stelle
giorno ora
risposta
id
utente_matricola
commento_id
testo
giorno
ora
ricevere
utente_matricola
notifica_id
giorno
ora
utente
matricola
nome
cognome
email
password
n_telefono
stato_ban
n_ban
presentazione
stato_presentazione
n_pubblicazioni
n_segnalazione_immotivate
giorno_iscrizione
ultimo_accesso
n_recensioni
n_like
n_stelle
n_commenti
livello_utenza
utente_bannato
utente_matricola
motivazione
giorno
ora
commento
id
recensione_id
utente_matricola
testo
giorno
ora
notifica
id
testo
tipo_entita
riferimento_entita
tipo_notifica
apprezzare
utente_matricola
recensione_id
stelle

QUERY

1. Modifica del livello di un utente da ATTIVO a PASSIVO

Nel nostro caso riconosciamo un utente passivo dallo stato di approvazione della presentazione che sarà messo a 0, mentre riconosceremo gli altri livelli di utenza dalla combinazione del flag stato_presentazione a 1 (che indicherà che si tratta di un utente attivo) e il flag livello utenza che indicherà i permessi che ha l'utente attivo (NULL = attivo, 0 = moderatore, 1 = gestore).

```
UPDATE utente SET stato_presentazione = 1 WHERE matricola =
“707287"
```

```
mysql> SELECT matricola, presentazione, stato_presentazione, livello_utenza FROM utente;
+-----------+---------------+---------------------+----------------+
| matricola | presentazione | stato_presentazione | livello_utenza |
+-----------+---------------+---------------------+----------------+
| 181020    | presentazione |                   1 |           NULL |
| 221919    | NULL          |                   0 |           NULL |
| 461442    | presentazione |                   1 |              1 |
| 707287    | presentazione |                   0 |           NULL |
| 864293    | presentazione |                   1 |              0 |
+-----------+---------------+---------------------+----------------+
```

```
mysql> UPDATE utente SET stato_presentazione = 1 WHERE matricola =  "707287";
Query OK, 1 row affected (0,00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT matricola, presentazione, stato_presentazione, livello_utenza FROM utente;
+-----------+---------------+---------------------+----------------+
| matricola | presentazione | stato_presentazione | livello_utenza |
+-----------+---------------+---------------------+----------------+
| 181020    | presentazione |                   1 |           NULL |
| 221919    | NULL          |                   0 |           NULL |
| 461442    | presentazione |                   1 |              1 |
| 707287    | presentazione |                   1 |           NULL |
| 864293    | presentazione |                   1 |              0 |
+-----------+---------------+---------------------+----------------+
```

2. Estrazione elenco delle ultime dieci pubblicazioni inserite

```
SELECT isbn FROM pubblicazione ORDER BY giorno_pubblicazione DESC
LIMIT 10;
```

Questa query, avendo riempito il database con sole 5 pubblicazioni ci darà in output

gli isbn di tutte le pubblicazioni presenti ma la verifica della correttezza è banale in quanto è una query molto semplice.

```
mysql> SELECT isbn FROM pubblicazione
    -> ORDER BY giorno_pubblicazione DESC LIMIT 10;
+---------------+
| isbn          |
+---------------+
| 7198090073652 |
| 9977757714223 |
| 5763622736558 |
| 4734401332680 |
| 1408892618492 |
+---------------+
```

3. Estrazione elenco delle pubblicazioni aggiornate di recente (ultimi 30 giorni).

Come aggiornamento intendiamo una eventuale modifica della pubblicazione e non l'aggiunta di una pubblicazione, perciò andremo a controllare in storia dove il campo azione sarà uguale a "modificare".

```
SELECT pubblicazione_isbn FROM storia
       WHERE azione = "modificare" && TIMESTAMPDIFF(DAY,
CURRENT_TIMESTAMP(), time_stamp) <= 30
       GROUP BY pubblicazione_isbn;
```

```
mysql> select * from storia;
+----+------------+---------------------+------------------+--------------------+
| id | azione     | time_stamp          | utente_matricola | pubblicazione_isbn |
+----+------------+---------------------+------------------+--------------------+
|  1 | aggiungere | 2019-07-04 16:18:07 | 461442           | 9977757714223      |
|  2 | eliminare  | 2019-07-04 16:18:07 | 181020           | 9977757714223      |
|  3 | aggiungere | 2019-07-04 16:18:07 | 461442           | 4734401332680      |
|  4 | aggiungere | 2019-07-04 16:18:07 | 864293           | 7198090073652      |
|  5 | aggiungere | 2019-07-04 16:18:07 | 221919           | 1408892618492      |
|  6 | modificare | 2019-07-04 16:18:07 | 461442           | 4734401332680      |
+----+------------+---------------------+------------------+--------------------+
6 rows in set (0,00 sec)

mysql> SELECT pubblicazione_isbn FROM storia
    -> WHERE azione = "modificare" &&
    -> TIMESTAMPDIFF(DAY, CURRENT_TIMESTAMP(), time_stamp) <= 30
    -> GROUP BY pubblicazione_isbn;
+--------------------+
| pubblicazione_isbn |
+--------------------+
| 4734401332680      |
+--------------------+
```

4. Estrazione elenco degli utenti più "collaborativi" (cioè quelli che hanno inserito più pubblicazioni).

Questa è una query interattiva che prenderà in ingresso il numero di utenti che si vuole avere in uscita, nell'esempio usiamo 3 come valore per testarne il funzionamento.

```
SELECT matricola FROM utente ORDER BY n_pubblicazioni DESC LIMIT 3
```

```
mysql> SELECT matricola, n_pubblicazioni FROM utente;
+-----------+-----------------+
| matricola | n_pubblicazioni |
+-----------+-----------------+
| 181020    |             272 |
| 221919    |            1388 |
| 461442    |             808 |
| 707287    |            1008 |
| 864293    |            1746 |
+-----------+-----------------+
5 rows in set (0,00 sec)

mysql> SELECT matricola FROM utente ORDER BY n_pubblicazioni DESC LIMIT 3;
+-----------+
| matricola |
+-----------+
| 864293    |
| 221919    |
| 707287    |
+-----------+
```

5. Estrazione elenco delle pubblicazioni inserite da un utente.

```
SELECT isbn FROM pubblicazione WHERE utente_matricola = "707287";
```

```
mysql> SELECT isbn, utente_matricola FROM pubblicazione;
+---------------+------------------+
| isbn          | utente_matricola |
+---------------+------------------+
| 1408892618492 | 181020           |
| 4734401332680 | 181020           |
| 9977757714223 | 461442           |
| 5763622736558 | 707287           |
| 7198090073652 | 707287           |
+---------------+------------------+
5 rows in set (0,00 sec)

mysql> SELECT isbn FROM pubblicazione WHERE utente_matricola = "707287";
+---------------+
| isbn          |
+---------------+
| 5763622736558 |
| 7198090073652 |
+---------------+
```

6. Estrazione catalogo, cioè elenco di tutte le pubblicazioni con titolo, autori, editore e anno di pubblicazione, ordinato per titolo.

Nel risultato di questa query vedremo il duplicarsi delle pubblicazioni, in quanto, per rendere il sistema più realistico, abbiamo considerato che una pubblicazione sia stata scritta da più autori perciò vedremo che ogni pubblicazione parteciperà al risultato finale in un numero di tuple pari al numero di autori che l'hanno scritta.

```
SELECT pubblicazione.titolo, pubblicazione.editore,
pubblicazione.giorno_pubblicazione, autore.nome, autore.cognome
FROM autore
        INNER JOIN scrivere ON autore.id = scrivere.autore_id
        INNER JOIN pubblicazione ON pubblicazione.isbn =
scrivere.pubblicazione_isbn
        ORDER BY pubblicazione.titolo;
```

```
mysql> SELECT pubblicazione.titolo, pubblicazione.editore, pubblicazione.giorno_pubblicazione,
    -> autore.nome, autore.cognome FROM autore
    ->          INNER JOIN scrivere ON autore.id = scrivere.autore_id
    ->          INNER JOIN pubblicazione ON pubblicazione.isbn = scrivere.pubblicazione_isbn
    ->          ORDER BY pubblicazione.titolo;
+--------+---------+----------------------+---------+---------+
| titolo | editore | giorno_pubblicazione | nome    | cognome |
+--------+---------+----------------------+---------+---------+
| atomic | King    | 2049-04-28           | Jessie  | Houston |
| atomic | King    | 2049-04-28           | Cecilia | Jacobs  |
| law    | Ortega  | 2101-11-05           | Cecilia | Jacobs  |
| mind   | Summers | 2107-02-05           | Jessie  | Houston |
| mind   | Summers | 2107-02-05           | Emily   | Peters  |
| mind   | Summers | 2107-02-05           | Leah    | Sharp   |
| sense  | Summers | 2116-01-10           | Leah    | Sharp   |
+--------+---------+----------------------+---------+---------+
```

7. Estrazione dati completi di una pubblicazione specifica dato il suo ID.

```
SELECT * FROM pubblicazione WHERE isbn = "1408892618492";
```

```
mysql> SELECT * FROM pubblicazione WHERE isbn = "1408892618492";
+---------------+----------------------+-------------------+----------+----------+
| isbn          | giorno_pubblicazione | ora_pubblicazione | lingua   | n_pagine |
+---------------+----------------------+-------------------+----------+----------+
| 1408892618492 | 2049-04-28           | 14:23:54          | italiano |      482 |
+---------------+----------------------+-------------------+----------+----------+
+--------+---------------+--------+---------+---------------------+------------------+
| titolo | descrizione   | indice | editore | stato_pubblicazione | utente_matricola |
+--------+---------------+--------+---------+---------------------+------------------+
| atomic | descrizione 2 | NULL   | King    |                   0 | 181020           |
+--------+---------------+--------+---------+---------------------+------------------+
```

8. Ricerca di pubblicazioni per ISBN, titolo, autore, e parole chiave.

Per questa query abbiamo pensato di realizzare quattro query distinte che filtrano rispettivamente per titolo, isbn, autore e parole chiave.

```
SELECT * FROM pubblicazione WHERE titolo = “atomic”;
```

```
mysql> select * from pubblicazione
    -> WHERE titolo = "atomic";
+---------------+----------------------+-------------------+----------+----------+
| isbn          | giorno_pubblicazione | ora_pubblicazione | lingua   | n_pagine |
+---------------+----------------------+-------------------+----------+----------+
| 1408892618492 | 2049-04-28           | 14:23:54          | italiano |      482 |
+---------------+----------------------+-------------------+----------+----------+

+--------+--------------+--------+---------+---------------------+------------------+
| titolo | descrizione  | indice | editore | stato_pubblicazione | utente_matricola |
+--------+--------------+--------+---------+---------------------+------------------+
| atomic | descrizione 2 | NULL   | King    |                   0 | 181020           |
+--------+--------------+--------+---------+---------------------+------------------+
```

```
SELECT * FROM pubblicazione WHERE isbn = 1408892618492;
```

```
mysql> SELECT * FROM pubblicazione WHERE isbn = 1408892618492;
+---------------+----------------------+-------------------+----------+----------+
| isbn          | giorno_pubblicazione | ora_pubblicazione | lingua   | n_pagine |
+---------------+----------------------+-------------------+----------+----------+
| 1408892618492 | 2049-04-28           | 14:23:54          | italiano |      482 |
+---------------+----------------------+-------------------+----------+----------+

+--------+--------------+--------+---------+---------------------+------------------+
| titolo | descrizione  | indice | editore | stato_pubblicazione | utente_matricola |
+--------+--------------+--------+---------+---------------------+------------------+
| atomic | descrizione 2 | NULL   | King    |                   0 | 181020           |
+--------+--------------+--------+---------+---------------------+------------------+
```

```
SELECT pubblicazione.isbn FROM autore
        INNER JOIN scrivere ON autore.id = scrivere.autore_id
        INNER JOIN pubblicazione
        ON pubblicazione.isbn = scrivere.pubblicazione_isbn
        WHERE id = “2";
```

```
mysql> SELECT pubblicazione.isbn FROM autore
    ->         INNER JOIN scrivere ON autore.id = scrivere.autore_id
    ->         INNER JOIN pubblicazione
    ->    ON pubblicazione.isbn = scrivere.pubblicazione_isbn
    ->         WHERE id = "2";
+---------------+
| isbn          |
+---------------+
| 7198090073652 |
| 9977757714223 |
+---------------+
```

```
SELECT pubblicazione_isbn FROM contenere
        WHERE parola_chiave = “classi";
```

```
mysql> select * from contenere;
+---------------+--------------------+
| parola_chiave | pubblicazione_isbn |
+---------------+--------------------+
| classi        | 1408892618492      |
| classi        | 5763622736558      |
| linguaggio    | 7198090073652      |
| classi        | 9977757714223      |
| ereditarietà  | 9977757714223      |
+---------------+--------------------+

mysql> SELECT pubblicazione_isbn FROM contenere
    ->         WHERE parola_chiave = "classi";
+--------------------+
| pubblicazione_isbn |
+--------------------+
| 1408892618492      |
| 5763622736558      |
| 9977757714223      |
+--------------------+
```

9. Inserimento di una recensione relativa a una pubblicazione.

In questa query è presente un vincolo, in quanto un utente non deve poter inserire più di una recensione per la stessa pubblicazione. Il seguente vincolo è stato realizzato in SQL tramite un CONSTRAINT nel seguente modo:

```
CREATE TABLE recensione (
    id int auto_increment,
    stato_recensione boolean NOT NULL default 0,    /* 0 = non
     accettata, 1 = accettata */
    testo text NOT NULL,
    stelle int NOT NULL CONSTRAINT stelle_ck CHECK(stelle >= 0 and
     stelle < 6),
    giorno date NOT NULL,
    ora time NOT NULL,
    utente_matricola varchar(10) NOT NULL,
    pubblicazione_isbn varchar(15) NOT NULL,
    foreign key(utente_matricola)references utente(matricola),
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    CONSTRAINT una_recensione_per_pubblicazione
     UNIQUE(utente_matricola, pubblicazione_isbn),
    primary key(id)
);
```

Andiamo a testare il funzionamento di questo vincolo :

Questa è la tabella recensioni

```
mysql> SELECT id, utente_matricola, pubblicazione_isbn FROM recensione;
+----+------------------+--------------------+
| id | utente_matricola | pubblicazione_isbn |
+----+------------------+--------------------+
|  2 | 221919           | 4734401332680      |
|  5 | 221919           | 5763622736558      |
|  1 | 221919           | 9977757714223      |
|  4 | 461442           | 1408892618492      |
|  3 | 707287           | 7198090073652      |
+----+------------------+--------------------+
```

Andiamo ora ad eseguire la stessa query di inserimento due volte. Se il funzionamento è corretto, la prima volta che andremo ad eseguire la query essa andrà a buon fine, mentre la seconda volta darà un errore dipeso dall'aggiunta di una seconda recensione da parte di un utente per la stessa pubblicazione.

La query che andremo a lanciare è la seguente.

```
INSERT INTO recensione(stato_recensione, testo, stelle, giorno,
ora, utente_matricola, pubblicazione_isbn) VALUES (0, "recensione
numero sei", 4, '2019/03/07', '11:11:11', 221919, 7198090073652);
```

Primo lancio della query

```
mysql> INSERT INTO recensione(stato_recensione, testo, stelle,
    -> giorno, ora, utente_matricola, pubblicazione_isbn)
    -> VALUES (0, "recensione numero sei", 4, '2019/03/07',
    -> '11:11:11', 221919, 7198090073652);
Query OK, 1 row affected (0,01 sec)

mysql> SELECT id, utente_matricola, pubblicazione_isbn FROM recensione;
+----+------------------+--------------------+
| id | utente_matricola | pubblicazione_isbn |
+----+------------------+--------------------+
|  2 | 221919           | 4734401332680      |
|  5 | 221919           | 5763622736558      |
|  9 | 221919           | 7198090073652      |
|  1 | 221919           | 9977757714223      |
|  4 | 461442           | 1408892618492      |
|  3 | 707287           | 7198090073652      |
+----+------------------+--------------------+
```

Come possiamo vedere la query è andata a buon fine e ne risulterà l'aggiunta della recensione con l'id 9.

Secondo lancio della query.

```
mysql> INSERT INTO recensione(stato_recensione, testo, stelle,
    -> giorno, ora, utente_matricola, pubblicazione_isbn)
    -> VALUES (0, "recensione numero sei", 4, '2019/03/07',
    -> '11:11:11', 221919, 7198090073652);
ERROR 1062 (23000): Duplicate entry '221919-7198090073652' for key 'una_recensione_per_pubblicazione'
```

Come volevasi dimostrare in output avremo l'errore sul constraint: 'una_recensione_per_pubblicazione'.

10. Approvazione di una recensione (da parte del moderatore).

Le recensioni hanno un flag di approvazione chiamato 'stato_recensione' (0 = non approvata, 1 = approvata).

```
mysql> SELECT id, stato_recensione, testo FROM recensione;
+----+------------------+--------------------------+
| id | stato_recensione | testo                    |
+----+------------------+--------------------------+
|  1 |                0 | recensione numero uno    |
|  2 |                0 | recensione numero due    |
|  3 |                1 | recensione numero tre    |
|  4 |                0 | recensione numero quattro |
|  5 |                1 | recensione numero cinque  |
|  9 |                0 | recensione numero sei    |
+----+------------------+--------------------------+
```

```
UPDATE recensione SET stato_recensione = 1 WHERE id = 1;
```

```
mysql> UPDATE recensione SET stato_recensione = 1 WHERE id = 1;
Query OK, 1 row affected (0,00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT id, stato_recensione, testo FROM recensione;
+----+------------------+--------------------------+
| id | stato_recensione | testo                    |
+----+------------------+--------------------------+
|  1 |                1 | recensione numero uno    |
|  2 |                0 | recensione numero due    |
|  3 |                1 | recensione numero tre    |
|  4 |                0 | recensione numero quattro |
|  5 |                1 | recensione numero cinque  |
|  9 |                0 | recensione numero sei    |
+----+------------------+--------------------------+
```

11. Inserimento di un like relativo a una pubblicazione.

```
mysql> select * from noje;
+--------------------+------------------+------------+----------+
| pubblicazione_isbn | utente_matricola | giorno     | ora      |
+--------------------+------------------+------------+----------+
| 1408892618492      | 181020           | 2091-03-30 | 22:11:34 |
| 1408892618492      | 864293           | 2050-11-12 | 12:23:43 |
| 5763622736558      | 181020           | 2069-10-10 | 11:22:33 |
| 7198090073652      | 461442           | 2067-12-06 | 01:23:21 |
| 9977757714223      | 221919           | 2086-12-06 | 21:21:32 |
+--------------------+------------------+------------+----------+
```

```
INSERT INTO noje VALUES ("1408892618492", "221919", CURDATE(),
NOW());
```

```
mysql> INSERT INTO noje(pubblicazione_isbn, utente_matricola, giorno, ora)
    -> values("1408892618492", "221919", CURDATE(), NOW());
Query OK, 1 row affected (0,00 sec)

mysql> select * from noje;
+--------------------+------------------+------------+----------+
| pubblicazione_isbn | utente_matricola | giorno     | ora      |
+--------------------+------------------+------------+----------+
| 1408892618492      | 181020           | 2091-03-30 | 22:11:34 |
| 1408892618492      | 221919           | 2019-07-04 | 22:26:41 |
| 1408892618492      | 864293           | 2050-11-12 | 12:23:43 |
| 5763622736558      | 181020           | 2069-10-10 | 11:22:33 |
| 7198090073652      | 461442           | 2067-12-06 | 01:23:21 |
| 9977757714223      | 221919           | 2086-12-06 | 21:21:32 |
+--------------------+------------------+------------+----------+
```

Per questa query vale lo stesso discorso della query numero 9 , ovvero che un utente può inserire un solo like per ogni pubblicazione. La soluzione che verrà adottata sarà identica e verrà illustrata di seguito.

Tabella Like con constraint 'un_like_per_pubblicazione'

```
CREATE TABLE noje (
    pubblicazione_isbn varchar(15),
    utente_matricola varchar(10),
    giorno date NOT NULL,
    ora time NOT NULL,
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    foreign key(utente_matricola)references utente(matricola),
    CONSTRAINT un_like_pre_pubblicazione
     UNIQUE(pubblicazione_isbn, utente_matricola),
    primary key(pubblicazione_isbn, utente_matricola)
);
```

12. Calcolo numero dei like per una pubblicazione.

```
mysql> select * from noje;
+--------------------+------------------+------------+----------+
| pubblicazione_isbn | utente_matricola | giorno     | ora      |
+--------------------+------------------+------------+----------+
| 1408892618492      | 181020           | 2091-03-30 | 22:11:34 |
| 1408892618492      | 864293           | 2050-11-12 | 12:23:43 |
| 5763622736558      | 181020           | 2069-10-10 | 11:22:33 |
| 7198090073652      | 461442           | 2067-12-06 | 01:23:21 |
| 9977757714223      | 221919           | 2086-12-06 | 21:21:32 |
+--------------------+------------------+------------+----------+
```

```
SELECT COUNT(pubblicazione) FROM noje WHERE pubblicazione_isbn =
"1408892618492";
```

```
mysql> SELECT COUNT(pubblicazione_isbn) FROM noje
    -> WHERE pubblicazione_isbn = "1408892618492";
+---------------------------+
| COUNT(pubblicazione_isbn) |
+---------------------------+
|                         2 |
+---------------------------+
```

13. Estrazione elenco delle recensioni approvate per una pubblicazione.

```
mysql> SELECT id, stato_recensione, pubblicazione_isbn FROM recensione;
+----+------------------+--------------------+
| id | stato_recensione | pubblicazione_isbn |
+----+------------------+--------------------+
|  1 |                0 | 9977757714223      |
|  2 |                0 | 4734401332680      |
|  3 |                1 | 7198090073652      |
|  4 |                0 | 1408892618492      |
|  5 |                1 | 5763622736558      |
+----+------------------+--------------------+
```

```
SELECT id FROM recensione
WHERE pubblicazione_isbn = "7198090073652" &&
     stato_recensione == 1;
```

```
mysql> SELECT id FROM recensione
    -> WHERE pubblicazione_isbn = "7198090073652"
    -> && stato_recensione = 1;
+----+
| id |
+----+
|  3 |
+----+
```

14. Estrazione elenco delle recensioni in attesa di approvazione.

```
SELECT id FROM recensione WHERE stato_recensione = 0;
```

```
mysql> SELECT id FROM recensione WHERE stato_recensione = 0;
+----+
| id |
+----+
|  1 |
|  2 |
|  4 |
+----+
```

15. Estrazione log delle modifiche effettuate su una pubblicazione.

Per log delle modifiche abbiamo inteso tutto ciò che risulta nella storia di quella pubblicazione (aggiunta, modifica ed eliminazione).

```
SELECT * FROM storia WHERE pubblicazione_isbn = "1408892618492";
```

```
mysql> SELECT * FROM storia WHERE pubblicazione_isbn = "1408892618492";
+----+------------+---------------------+------------------+--------------------+
| id | azione     | time_stamp          | utente_matricola | pubblicazione_isbn |
+----+------------+---------------------+------------------+--------------------+
|  5 | aggiungere | 2019-07-04 22:49:56 | 221919           | 1408892618492      |
+----+------------+---------------------+------------------+--------------------+
```

16. Estrazione elenco delle pubblicazioni per le quali è disponibile un download.

```
mysql> SELECT id, tipo, pubblicazione_isbn FROM sorgente;
+----+----------+--------------------+
| id | tipo     | pubblicazione_isbn |
+----+----------+--------------------+
|  1 | immagine | 5763622736558      |
|  2 | download | 1408892618492      |
|  3 | acquisto | 7198090073652      |
|  4 | immagine | 4734401332680      |
|  5 | video    | 9977757714223      |
+----+----------+--------------------+
```

```
SELECT pubblicazione_isbn FROM sorgente WHERE tipo = "download" ;
```

```
mysql> SELECT pubblicazione_isbn FROM sorgente WHERE tipo = "download" ;
+--------------------+
| pubblicazione_isbn |
+--------------------+
| 1408892618492      |
+--------------------+
```

17. Estrazione della lista delle pubblicazioni in catalogo, ognuna con la data dell'ultima ristampa.

```
SELECT pubblicazione_isbn, giorno FROM ristampa;
```

```
mysql> SELECT pubblicazione_isbn, giorno FROM ristampa;
+--------------------+------------+
| pubblicazione_isbn | giorno     |
+--------------------+------------+
| 5763622736558      | 2011-10-25 |
| 4734401332680      | 2018-12-30 |
| 9977757714223      | 2017-06-12 |
| 5763622736558      | 2018-09-24 |
| 5763622736558      | 2012-03-23 |
+--------------------+------------+
```

18. Data una pubblicazione, restituire tutte le pubblicazioni del catalogo aventi gli stessi autori.

```
mysql> SELECT * FROM autore;
+----+----------+---------+
| id | nome     | cognome |
+----+----------+---------+
|  1 | Emily    | Peters  |
|  2 | Leah     | Sharp   |
|  3 | Cecilia  | Jacobs  |
|  4 | Jessie   | Houston |
|  5 | Beatrice | Gill    |
+----+----------+---------+
```

```
mysql> SELECT isbn, titolo FROM pubblicazione;
+---------------+--------+
| isbn          | titolo |
+---------------+--------+
| 1408892618492 | atomic |
| 4734401332680 | sense  |
| 5763622736558 | law    |
| 7198090073652 | sense  |
| 9977757714223 | mind   |
+---------------+--------+
```

```
mysql> SELECT * FROM scrivere;
+-----------+-------------------+
| autore_id | pubblicazione_isbn |
+-----------+-------------------+
|         3 | 1408892618492      |
|         4 | 1408892618492      |
|         3 | 5763622736558      |
|         2 | 7198090073652      |
|         1 | 9977757714223      |
|         2 | 9977757714223      |
|         4 | 9977757714223      |
+-----------+-------------------+
```

```
SELECT i.pubblicazione_isbn FROM scrivere i
INNER JOIN scrivere ON i.autore_id = scrivere.autore_id
INNER JOIN  scrivere u ON i.autore_id = u.autore_id
WHERE u.pubblicazione_isbn = 9977757714223
GROUP BY i.pubblicazione_isbn;
```

```
mysql> SELECT i.pubblicazione_isbn FROM scrivere i
    -> INNER JOIN scrivere ON i.autore_id = scrivere.autore_id
    -> INNER JOIN  scrivere u ON i.autore_id = u.autore_id
    -> WHERE u.pubblicazione_isbn = 9977757714223
    -> GROUP BY i.pubblicazione_isbn;
+--------------------+
| pubblicazione_isbn |
+--------------------+
| 9977757714223      |
| 7198090073652      |
| 1408892618492      |
+--------------------+
```

QUERY AGGIUNTIVE

A. Soft ban di un utente

Quando un utente bannato (soft), la data del suo ultimo accesso sarà settata alla data corrente così sarà questo campo a determinare il passaggio dei 30 giorni. Inoltre il flag n_ban sarà incrementato e il flag stato_ban sarà messo a 0 (0 = soft ban).

```
UPDATE utente SET n_ban = n_ban + 1 WHERE matricola = "864293";
UPDATE utente SET stato_ban = 0 WHERE matricola = "864293";
UPDATE utente SET ultimo_accesso = NOW() WHERE matricola =
“864293";
```

B. Controllo accesso utente

Quando un utente effettua l'accesso al sistema si dovranno controllare i seguenti parametri.

Controllo se l'utente è bannato (il controllo verrà fatto in java a partire dai dati risultanti da questa query)

```
SELECT stato_ban, n_ban, ultimo_accesso from utente WHERE
matricola = “864293";
```

Query che riattiva l'utente dopo uno stato di soft ban se sono passati 30 giorni

```
UPDATE utente SET stato_ban = null WHERE matricola = "864293";
```

Query che banna permanentemente un utente nel caso sia arrivato a tre soft ban

```
UPDATE utente SET stato_ban = 1 WHERE matricola = “864293";
```

Salvataggio della matricola dell'utente bannato all'interno della tabella utente_bannato insieme alla motivazione, alla data e all'ora del ban

```
INSERT INTO utente_bannato(utente_matricola, motivazione, giorno,
ora) values("864293", "motivo", CURDATE(), NOW());
```

Query che restituisce il numero di segnalazioni immotivate fatte dall'utente, nel caso in cui siamo uguali a 3 verrà lanciato un soft ban tramite le query riportate precedentemente e verrà rimesso a 0 il contatore del numero di segnalazioni

Numero segnalazioni immotivate

```
SELECT n_segnalazioni_immotivate FROM utente WHERE matricola =
"864293";
```

Rettare a zero il contatore

```
UPDATE utente SET n_segnalazioni_immotivate = 0 WHERE matricola =
"864293";
```

C. Incremento segnalazioni immotivate utente

```
UPDATE utente SET n_segnalazioni_immotivate =
n_segnalazioni_immotivate + 1 WHERE matricola = "221919";
```

D. Query per passaggio da utente attivo a moderatore

La prima cosa da controllare è se un utente sia attivo, questo verrà fatto con la seguente query.

```
SELECT stato_presentazione FROM utente WHERE matricola = "221919";
```

Query che restituisce la lista degli utenti che non sono mai stati bannati.

```
SELECT matricola from utente WHERE n_ban = 0;
```

Query che restituisce il numero di ban di un dato utente.

```
SELECT n_ban from utente WHERE matricola = "221919";
```

Passaggio di un utente da attivo a moderatore

```
UPDATE utente SET livello_utenza = 0 WHERE matricola = "221919";
```

E. Modifica di un commento inappropriato da parte di un moderatore

Questa query è divisa in due fasi. La prima consiste nel restituire il testo del commento che il moderatore modificherà tramite l'interfaccia, mentre la seconda prenderà in input la matricola del moderatore che ha effettuato la modifica ed il nuovo testo ed aggiornerà sia il testo del commento sia la matricola dell'utente che lo ha rilasciato.

Query che restituisce il testo del commento

```
SELECT testo FROM commento WHERE id = 5;
```

Aggiornamento testo del commento

```
UPDATE commento SET testo = "nuovo testo 5" WHERE id = 5;
```

Aggiornamento utente che ha rilasciato il commento

```
UPDATE commento SET utente_matricola = "864293" WHERE id = 5;
```

F. Apprezzamento di una recensione

Nel campo stelle relativo all'apprezzamento di una recensione è previsto un vincolo che prevede l'inserimento di un numero di stelle compreso tra 0 e 5. Lanceremo due query che andranno rispettivamente a dare un grado di apprezzamento di due e sei stelle per una recensione per testare il corretto funzionamento del vincolo.

```
INSERT INTO apprezzare(utente_matricola, recensione_id, stelle)
values("221919", 1, 2);
```

```
mysql> INSERT INTO apprezzare(utente_matricola, recensione_id, stelle) values("221919", 1, 2);
Query OK, 1 row affected (0,01 sec)

mysql> SELECT * FROM apprezzare;
+------------------+---------------+--------+
| utente_matricola | recensione_id | stelle |
+------------------+---------------+--------+
| 221919           |             1 |      2 |
+------------------+---------------+--------+
```

```
INSERT INTO apprezzare(utente_matricola, recensione_id, stelle)
values("221919", 2, 6);
```

```
mysql> INSERT INTO apprezzare(utente_matricola, recensione_id, stelle) values("221919", 2, 6);
ERROR 3819 (HY000): Check constraint 'stelle_ck2' is violated.
```

G. Segnalazione di un commento

```
INSERT INTO notifica(tipo_notifica, tipo_entita,
riferimento_entita, testo) values("segnalazione", "commento", "5",
"commento non appropriato");
```

H. Eliminazione di una pubblicazione

Questa operazione comporta l'eliminazione di tutto ciò che è legato alla pubblicazione che viene eliminata. Essendo una query aggiunta da noi consideriamo l'eliminazione dei commenti, risposte e e recensioni legate alla pubblicazione eliminata.

Con questa query prendiamo gli id dei commenti associati alle relazioni in modo da andare ad eliminare le risposte tramite questo id.

```
SELECT c.id FROM commento c
INNER JOIN recensione r ON r.id = c.recensione_id
WHERE r.pubblicazione_isbn = "1408892618492";
```

Eliminazione delle risposte associate agli id calcolati nella query appena descritta.

```
DELETE FROM risposta WHERE commento_id = (
        SELECT c.id FROM commento c
        INNER JOIN recensione r ON r.id = c.recensione_id
        WHERE r.pubblicazione_isbn = "1408892618492"
        );
```

Query per calcolare gli id delle recensioni associate alla pubblicazione da eliminare in modo da cancellare i commenti legati a questo id.

```
SELECT id FROM recensione WHERE pubblicazione_isbn =
"1408892618492";
```

Eliminazione dei commenti

```
DELETE FROM commento WHERE recensione_id = (
        SELECT id FROM recensione
        WHERE pubblicazione_isbn = "1408892618492"
        );
```

Eliminazione delle recensioni

```
DELETE FROM recensione WHERE pubblicazione_isbn = "1408892618492";
```

Eliminazione della pubblicazione

```
DELETE FROM pubblicazione WHERE isbn = "1408892618492";
```

SQL
CREAZIONE TABELLE

```
/* Tabelle Entità e Relazioni */

CREATE TABLE utente (
    matricola varchar(10),
    nome varchar(20) NOT NULL,
    cognome varchar(20) NOT NULL,
    email varchar(59) NOT NULL unique,
    password varchar(30) NOT NULL CONSTRAINT password_ck
     CHECK(char_length(password) > 7),
    n_telefono varchar(10),
    stato_ban boolean default null,      /* null = normale, 0 =
     soft_ban | 1 = perma_ban */
    n_ban int NOT NULL default 0,
    ultimo_accesso date NOT NULL,
    giorno_iscrizione date NOT NULL,
    n_commenti int NOT NULL default 0 CONSTRAINT n_commenti_ck
     CHECK(n_commenti >= 0),
    n_recensioni int NOT NULL default 0 CONSTRAINT n_recensioni_ck
     CHECK(n_recensioni >= 0),
    n_stelle int NOT NULL default 0 CONSTRAINT n_stelle_ck
     CHECK(n_stelle >= 0),
    n_pubblicazioni int NOT NULL default 0 CONSTRAINT
     n_pubblicazioni_ck CHECK(n_pubblicazioni >= 0),
    presentazione text default null,
    stato_presentazione boolean NOT NULL default 0,  /* 0 = non
     accettata, 1 = accettata */
    livello_utenza boolean default null,   /* null = attivo, 0 =
     moderatore, 1 = gestore ovviamente se u */
    n_segnalazioni_immotivate int default 0 NOT NULL CONSTRAINT
     segnalazioni_min_0_max_3 CHECK(n_segnalazioni_immotivate <= 3
     && n_segnalazioni_immotivate >= 0),
    primary key(matricola)
);
```

```
CREATE TABLE pubblicazione (
    isbn varchar(15),
    giorno_pubblicazione date NOT NULL,
    lingua enum('italiano', 'inglese', 'altro') NOT NULL,
    n_pagine int NOT NULL CONSTRAINT n_pagine_ck CHECK(n_pagine >
     0),
    titolo varchar(100) NOT NULL,
    descrizione text,
    indice blob,
    editore varchar(50) NOT NULL,
    stato_pubblicazione boolean NOT NULL default 0,    /* 0 = non
     accettata, 1 = accettata */
    utente_matricola varchar(10) NOT NULL,
    foreign key(utente_matricola)references utente(matricola),
    primary key(isbn)
);

CREATE TABLE sorgente (
    id int auto_increment,
    uri varchar(2100) NOT NULL,
    tipo enum('immagine', 'download', 'acquisto', 'video',
     'altro') NOT NULL,
    formato enum('jpg', 'pdf', 'epub', 'copertina rigida',
     'copertina flessibile', 'youtube', 'vimeo', 'altro'),
    descrizione text,
    pubblicazione_isbn varchar(15) NOT NULL,
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    primary key(id)
);
```

```
CREATE TABLE materia (
    nome varchar(20),
    primary key(nome)
);

CREATE TABLE ristampa (
    id int auto_increment,
    ora time NOT NULL,
    giorno date NOT NULL,
    numero int NOT NULL CONSTRAINT numero_ck CHECK(numero > 0),
    pubblicazione_isbn varchar(15) NOT NULL,
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    CONSTRAINT no_duplicati_numero_ristampa UNIQUE(numero,
     pubblicazione_isbn),
    primary key(id)
);

CREATE TABLE recensione (
    id int auto_increment,
    stato_recensione boolean NOT NULL default 0,    /* 0 = non
     accettata, 1 = accettata */
    testo text NOT NULL,
    stelle int NOT NULL CONSTRAINT stelle_ck CHECK(stelle >= 0 and
     stelle < 6),
    giorno date NOT NULL,
    ora time NOT NULL,
    utente_matricola varchar(10) NOT NULL,
    pubblicazione_isbn varchar(15) NOT NULL,
    foreign key(utente_matricola)references utente(matricola),
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    CONSTRAINT una_recensione_per_pubblicazione
     UNIQUE(utente_matricola, pubblicazione_isbn),
    primary key(id)
);
```

```
CREATE TABLE commento (
    id int auto_increment,
    testo text NOT NULL,
    giorno date NOT NULL,
    ora time NOT NULL,
    recensione_id int NOT NULL,
    utente_matricola varchar(10) NOT NULL,
    foreign key(recensione_id)references recensione(id),
    foreign key(utente_matricola)references utente(matricola),
    primary key(id)
);

CREATE TABLE risposta (
    id int auto_increment,
    testo text NOT NULL,
    giorno date NOT NULL,
    ora time NOT NULL,
    commento_id int NOT NULL,
    utente_matricola varchar(10) NOT NULL,
    foreign key(commento_id)references commento(id),
    foreign key(utente_matricola)references utente(matricola),
    primary key(id)
);

CREATE TABLE notifica (
    id int auto_increment,
    tipo_notifica enum('segnalazione', 'approvazione') NOT NULL,
    tipo_entita enum('commento', 'risposta', 'recensione',
     'pubblicazione', 'presentazione') NOT NULL,
    riferimento_entita varchar(50) NOT NULL, /* avrà un formato
     del tipo "id:{key}" */
    testo text NOT NULL,
    primary key(id)
);
```

```
CREATE TABLE utente_bannato (
    utente_matricola varchar(10),
    motivazione text NOT NULL,
    giorno date NOT NULL,
    ora time NOT NULL,
    foreign key(utente_matricola)references utente(matricola),
    primary key(utente_matricola)
);

CREATE TABLE storia (
    id int auto_increment,
    azione enum('aggiungere', 'eliminare', 'modificare') NOT NULL,
    time_stamp timestamp NOT NULL,
    utente_matricola varchar(10) NOT NULL,
    pubblicazione_isbn varchar(15) NOT NULL,
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    foreign key(utente_matricola)references utente(matricola),
    primary key(id)
);

CREATE TABLE autore (
    id int auto_increment,
    nome varchar(20) NOT NULL,
    cognome varchar(20) NOT NULL,
    primary key(id)
);

CREATE TABLE parola_chiave (
    parola varchar(20),
    primary key(parola)
);
```

```
CREATE TABLE contenere (
    parola_chiave varchar(20),
    pubblicazione_isbn varchar(20),
    foreign key (parola_chiave) references parola_chiave(parola),
    foreign key (pubblicazione_isbn) references
     pubblicazione(isbn),
    primary key (parola_chiave, pubblicazione_isbn)
);

CREATE TABLE scrivere (
    autore_id int,
    pubblicazione_isbn varchar(15),
    foreign key(autore_id)references autore(id),
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    primary key(autore_id, pubblicazione_isbn)
);

/* questa scelta è stata fatta per non avere valori nulli */
CREATE TABLE aggiornare (
    pubblicazione_isbn varchar(15) NOT NULL,
    riedizione_isbn varchar(15) NOT NULL unique,
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    foreign key(riedizione_isbn)references pubblicazione(isbn),
    primary key(riedizione_isbn)
);

CREATE TABLE appartenere (
    materia_nome varchar(20),
    pubblicazione_isbn varchar(15),
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    foreign key(materia_nome)references materia(nome),
    primary key(materia_nome, pubblicazione_isbn)
);
```

```
CREATE TABLE apprezzare (
    utente_matricola varchar(10),
    recensione_id int,
    stelle int NOT NULL CONSTRAINT stelle_ck2 CHECK(stelle >= 0
     and stelle < 6),
    foreign key(recensione_id)references recensione(id),
    foreign key(utente_matricola)references utente(matricola),
    primary key(recensione_id, utente_matricola)
);

CREATE TABLE noje (
    pubblicazione_isbn varchar(15),
    utente_matricola varchar(10),
    giorno date NOT NULL,
    ora time NOT NULL,
    foreign key(pubblicazione_isbn)references pubblicazione(isbn),
    foreign key(utente_matricola)references utente(matricola),
    CONSTRAINT un_like_pre_pubblicazione
     UNIQUE(pubblicazione_isbn, utente_matricola),
    primary key(pubblicazione_isbn, utente_matricola)
);

CREATE TABLE ricevere (
    utente_matricola varchar(10),
    notifica_id int,
    giorno date NOT NULL,
    ora time NOT NULL,
    foreign key(utente_matricola)references utente(matricola),
    foreign key(notifica_id)references notifica(id),
    primary key(utente_matricola, notifica_id)
);
```

SQL
INSERIMENTI

```
INSERT INTO utente(matricola, nome, cognome, email, password,
n_telefono, stato_ban, n_ban, ultimo_accesso, giorno_iscrizione,
n_commenti, n_recensioni, n_stelle, n_pubblicazioni,
stato_presentazione                , presentazione)
values( 707287, 'Lucinda', 'Knight', 'pukhujus@loul.gt',
'vSRl3kqkefsJ0', 3558654417, null, 0, '2001/05/22', '2001/11/05',
222, 1794, 1999, 1008, 0, "presentazione");

INSERT INTO utente(matricola, nome, cognome, email, password,
n_telefono, stato_ban, n_ban, ultimo_accesso, giorno_iscrizione,
n_commenti, n_recensioni, n_stelle, n_pubblicazioni,
stato_presentazione, livello_utenza, presentazione)
values( 461442, 'Dollie', 'Spencer', 'uhouvje@setrogha.pr',
'mmJYeysVr0MXIb4ucw', 3746145460, 0, 2, '2012/11/02',
'2013/09/21', 1841, 405, 466, 808, 1, 1, "presentazione");

INSERT INTO utente(matricola, nome, cognome, email, password,
n_telefono, stato_ban, n_ban, ultimo_accesso, giorno_iscrizione,
n_commenti, n_recensioni, n_stelle, n_pubblicazioni,
stato_presentazione                ) values( 221919, 'Lela',
'Sandoval', 'wa@je.it', 'Bkkc9PahJ0OZPOCSkS8', 3530173436, null,
2, '2011/06/19', '2042/06/30', 88, 810, 600, 1388, 0);

INSERT INTO utente(matricola, nome, cognome, email, password,
n_telefono, stato_ban, n_ban, ultimo_accesso, giorno_iscrizione,
n_commenti, n_recensioni, n_stelle, n_pubblicazioni,
stato_presentazione, livello_utenza, presentazione)
values( 864293, 'Victoria', 'Phillips', 'sodcu@ne.mg',
'SMm4bqFryhZjN', 3283061196, null, 2, '2016/04/23', '2018/09/29',
660, 320, 1903, 1746, 1, 0, "presentazione");

INSERT INTO utente(matricola, nome, cognome, email, password,
n_telefono, stato_ban, n_ban, ultimo_accesso, giorno_iscrizione,
n_commenti, n_recensioni, n_stelle, n_pubblicazioni,
stato_presentazione                , presentazione)
values( 181020, 'Lee', 'Chandler', 'emlaczo@zemje.lc',
'BekD4r2v4whsrhr', 3104390335, 1, 3, '2012/05/26', '2016/06/12',
792, 1624, 1967, 272, 1, "presentazione");
```

```
INSERT INTO pubblicazione(isbn, giorno_pubblicazione, lingua,
n_pagine, titolo, descrizione, editore, stato_pubblicazione,
utente_matricola) values(5763622736558, '2001/11/05', 'italiano',
458, 'law', 'descrizione 1', 'Ortega', 1, 707287);

INSERT INTO pubblicazione(isbn, giorno_pubblicazione, lingua,
n_pagine, titolo, descrizione, editore, stato_pubblicazione,
utente_matricola) values(1408892618492, '2019/04/28', 'italiano',
482, 'atomic', 'descrizione 2', 'King', 0, 181020);

INSERT INTO pubblicazione(isbn, giorno_pubblicazione, lingua,
n_pagine, titolo, descrizione, editore, stato_pubblicazione,
utente_matricola) values(7198090073652, '2016/01/10', 'inglese',
1733, 'sense', 'descrizione 3', 'Summers', 1, 707287);

INSERT INTO pubblicazione(isbn, giorno_pubblicazione, lingua,
n_pagine, titolo, descrizione, editore, stato_pubblicazione,
utente_matricola) values(4734401332680, '2012/08/27', 'italiano',
322, 'sense', 'descrizione 4',  'Summers', 0, 181020);

INSERT INTO pubblicazione(isbn, giorno_pubblicazione, lingua,
n_pagine, titolo, descrizione, editore, stato_pubblicazione,
utente_matricola) values(9977757714223, '2007/02/05', 'italiano',
839, 'mind', 'descrizione 5', 'Summers', 0, 461442);

INSERT INTO sorgente(uri, tipo, formato, descrizione,
pubblicazione_isbn) values( 'https://provauriIxEdVhb', 'immagine',
'jpg', 'descrizione 1', 5763622736558);

INSERT INTO sorgente(uri, tipo, formato, descrizione,
pubblicazione_isbn) values( 'https://provauriGjkTUKFcaAxbfgAZoQa',
'download', 'pdf', 'descrizione 2', 1408892618492);

INSERT INTO sorgente(uri, tipo, formato, descrizione,
pubblicazione_isbn) values( 'https://provauriUjPdJlyLhdq',
'acquisto', 'copertina flessibile', 'descrizione 3',
7198090073652);
```

```
INSERT INTO sorgente(uri, tipo, formato, descrizione,
pubblicazione_isbn) values( 'https://provauriMghZUJdQxyAye',
'immagine', 'jpg', 'descrizione 4', 4734401332680);

INSERT INTO sorgente(uri, tipo, formato, descrizione,
pubblicazione_isbn) values( 'https://provauriwYxGmfnN', 'video',
'youtube', 'descrizione 5', 9977757714223);

INSERT INTO materia(nome) values('informatica');
INSERT INTO materia(nome) values('biologia');
INSERT INTO materia(nome) values('fisica');
INSERT INTO materia(nome) values('biotecnologia');
INSERT INTO materia(nome) values('matematica');

INSERT INTO ristampa(pubblicazione_isbn, ora, giorno, numero)
values(5763622736558, '03:02:00', '2011/10/25', 1);

INSERT INTO ristampa(pubblicazione_isbn, ora, giorno, numero)
values(4734401332680, '11:04:54', '2018/12/30', 1);

INSERT INTO ristampa(pubblicazione_isbn, ora, giorno, numero)
values(9977757714223, '21:02:12', '2017/6/12/', 1);

INSERT INTO ristampa(pubblicazione_isbn, ora, giorno, numero)
values(5763622736558 , '07:23:24', '2018/9/24/', 2);

INSERT INTO ristampa(pubblicazione_isbn, ora, giorno, numero)
values(5763622736558 , '00:00:00', '2012/3/23/', 3);

INSERT INTO recensione(stato_recensione, testo, stelle, giorno,
ora, pubblicazione_isbn, utente_matricola) values(0, 'recensione
numero uno', 2, '2018/09/05', '11:03:12', 9977757714223, 221919);

INSERT INTO recensione(stato_recensione, testo, stelle, giorno,
ora, pubblicazione_isbn, utente_matricola) values(0, 'recensione
numero due', 0, '2013/11/08', '12:34:42', 4734401332680, 221919);
```

```
INSERT INTO recensione(stato_recensione, testo, stelle, giorno,
ora, pubblicazione_isbn, utente_matricola) values(1, 'recensione
numero tre', 2, '2006/08/05', '02:12:32', 7198090073652, 707287);

INSERT INTO recensione(stato_recensione, testo, stelle, giorno,
ora, pubblicazione_isbn, utente_matricola) values(0, 'recensione
numero quattro', 0, '2005/05/11', '21:12:21', 1408892618492,
461442);

INSERT INTO recensione(stato_recensione, testo, stelle, giorno,
ora, pubblicazione_isbn, utente_matricola) values(1, 'recensione
numero cinque', 3, '2005/04/05', '22:32:21', 5763622736558,
221919);

INSERT INTO utente_bannato(utente_matricola, motivazione, giorno,
ora) values( 181020, 'motivazione del ban', '2015/03/28',
'21:21:00');

INSERT INTO storia(azione, time_stamp, utente_matricola,
pubblicazione_isbn) values( 'aggiungere', CURRENT_TIMESTAMP(),
461442, 9977757714223);

INSERT INTO storia(azione, time_stamp, utente_matricola,
pubblicazione_isbn) values( 'eliminare', CURRENT_TIMESTAMP(),
181020, 9977757714223);

INSERT INTO storia(azione, time_stamp, utente_matricola,
pubblicazione_isbn) values( 'aggiungere', CURRENT_TIMESTAMP(),
461442, 4734401332680);

INSERT INTO storia(azione, time_stamp, utente_matricola,
pubblicazione_isbn) values( 'aggiungere', CURRENT_TIMESTAMP(),
864293, 7198090073652);
```

```
INSERT INTO storia(azione, time_stamp, utente_matricola,
pubblicazione_isbn) values( 'aggiungere', CURRENT_TIMESTAMP(),
221919, 1408892618492);

INSERT INTO storia(azione, time_stamp, utente_matricola,
pubblicazione_isbn) values( 'modificare', CURRENT_TIMESTAMP(),
461442, 4734401332680);

INSERT INTO autore(nome, cognome) values('Emily', 'Peters');
INSERT INTO autore(nome, cognome) values('Leah', 'Sharp');
INSERT INTO autore(nome, cognome) values('Cecilia', 'Jacobs');
INSERT INTO autore(nome, cognome) values('Jessie', 'Houston');
INSERT INTO autore(nome, cognome) values('Beatrice', 'Gill');

INSERT INTO parola_chiave(parola) values('ereditarietà');
INSERT INTO parola_chiave(parola) values('classi');
INSERT INTO parola_chiave(parola) values('linguaggio');
INSERT INTO parola_chiave(parola) values('automa');

INSERT INTO contenere(parola_chiave, pubblicazione_isbn)
values('ereditarietà', 9977757714223);

INSERT INTO contenere(parola_chiave, pubblicazione_isbn)
values('classi', 9977757714223);

INSERT INTO contenere(parola_chiave, pubblicazione_isbn)
values('linguaggio', 7198090073652);

INSERT INTO contenere(parola_chiave, pubblicazione_isbn)
values('classi', 1408892618492);

INSERT INTO contenere(parola_chiave, pubblicazione_isbn)
values('classi', 5763622736558);

INSERT INTO aggiornare(pubblicazione_isbn, riedizione_isbn)
values( 9977757714223, 4734401332680);
```

```
INSERT INTO aggiornare(pubblicazione_isbn, riedizione_isbn)
values( 4734401332680, 1408892618492);

INSERT INTO appartenere(materia_nome, pubblicazione_isbn)
values( 'informatica', 9977757714223);

INSERT INTO appartenere(materia_nome, pubblicazione_isbn)
values( 'informatica', 4734401332680);

INSERT INTO appartenere(materia_nome, pubblicazione_isbn)
values( 'biologia', 7198090073652);

INSERT INTO appartenere(materia_nome, pubblicazione_isbn)
values( 'fisica', 1408892618492);

INSERT INTO appartenere(materia_nome, pubblicazione_isbn)
values( 'matematica', 5763622736558);

INSERT INTO noje(pubblicazione_isbn, utente_matricola, giorno,
ora) values( 5763622736558, 181020, '2018/10/10' , '11:22:33');

INSERT INTO noje(pubblicazione_isbn, utente_matricola, giorno,
ora) values( 1408892618492, 181020, '2011/03/30' , '22:11:34');

INSERT INTO noje(pubblicazione_isbn, utente_matricola, giorno,
ora) values( 1408892618492, 864293, '2000/11/12' , '12:23:43');

INSERT INTO noje(pubblicazione_isbn, utente_matricola, giorno,
ora) values( 9977757714223, 221919, '2016/12/06' , '21:21:32');

INSERT INTO noje(pubblicazione_isbn, utente_matricola, giorno,
ora) values( 7198090073652, 461442, '2017/12/06' , '01:23:21');

INSERT INTO scrivere(autore_id, pubblicazione_isbn) values(1,
9977757714223);

INSERT INTO scrivere(autore_id, pubblicazione_isbn) values(2,
7198090073652);
```

```
INSERT INTO scrivere(autore_id, pubblicazione_isbn) values(2,
9977757714223);

INSERT INTO scrivere(autore_id, pubblicazione_isbn) values(3,
1408892618492);

INSERT INTO scrivere(autore_id, pubblicazione_isbn) values(3,
5763622736558);

INSERT INTO scrivere(autore_id, pubblicazione_isbn) values(4,
1408892618492);

INSERT INTO scrivere(autore_id, pubblicazione_isbn) values(4,
9977757714223);

INSERT INTO commento(testo, giorno, ora, recensione_id,
utente_matricola) values("commento 1", CURDATE(), NOW(), 1,
221919);

INSERT INTO commento(testo, giorno, ora, recensione_id,
utente_matricola) values("commento 2", CURDATE(), NOW(), 1,
221919);

INSERT INTO commento(testo, giorno, ora, recensione_id,
utente_matricola) values("commento 3", CURDATE(), NOW(), 3,
707287);

INSERT INTO commento(testo, giorno, ora, recensione_id,
utente_matricola) values("commento 4", CURDATE(), NOW(), 5,
181020);

INSERT INTO commento(testo, giorno, ora, recensione_id,
utente_matricola) values("commento 5", CURDATE(), NOW(), 4,
461442);
```

```
INSERT INTO risposta(testo, giorno, ora, commento_id,
utente_matricola) values("risposta 1", CURDATE(), NOW(), 1,
707287);

INSERT INTO risposta(testo, giorno, ora, commento_id,
utente_matricola) values("risposta 2", CURDATE(), NOW(), 2,
461442);

INSERT INTO risposta(testo, giorno, ora, commento_id,
utente_matricola) values("risposta 3", CURDATE(), NOW(), 3,
181020);

INSERT INTO risposta(testo, giorno, ora, commento_id,
utente_matricola) values("risposta 4", CURDATE(), NOW(), 4,
461442);

INSERT INTO risposta(testo, giorno, ora, commento_id,
utente_matricola) values("risposta 5", CURDATE(), NOW(), 5,
181020);

INSERT INTO risposta(testo, giorno, ora, commento_id,
utente_matricola) values("risposta 6", CURDATE(), NOW(), 4,
181020);
```