

# BASI DI DATI

## PROGETTO DI LABORATORIO

### PROGETTAZIONE

DIPARTIMENTO DI INFORMATICA

ANNO 2017/18

SILVESTRO STEFANO FRISULLO 832813 [\\_silvestro.frisull@edu.unito.it](mailto:silvestro.frisull@edu.unito.it)

ANTONINO BUSHAJ 847013 [antonino.bushaj@edu.unito.it](mailto:antonino.bushaj@edu.unito.it)

ALDO BUSHAJ 847091 [\\_aldo.bushaj@edu.unito.it](mailto:aldo.bushaj@edu.unito.it)

MIKELIAN GAZULLI 836089 [mikelian.gazulli@edu.unito.it](mailto:mikelian.gazulli@edu.unito.it)

## Requisiti Iniziali

Per gli utenti della piattaforma si memorizzano le informazioni richieste durante la registrazione. Come si ricava dal file *screenshot.pdf* nella sezione *utente*, per ogni utente si tiene traccia delle seguenti informazioni:

- libri scritti
- utenti seguiti
- utenti che seguono l'utente attuale
- lista dei libri attualmente in lettura
- si possono ignorare le conversazioni (requisito non richiesto)

I libri possono essere inseriti in diverse categorie:

- la lista delle categorie narrative e' visibile cliccando sul pulsante *scopri* in alto a sinistra, ogni libro viene inserito in almeno una categoria narrativa
- ogni libro può essere inserito anche in una delle categorie *primo piano*, *in salita* e *nuovo*

Ad ogni categoria narrativa vengono associati un insieme di scrittori correlati composto da tutti gli scrittori che hanno scritto almeno 2 libri nella categoria.

Nel file *screenshot.pdf* nella sezione *libro* sono visibili le informazioni associate a ciascun libro. I libri sono suddivisi in capitoli i cui titoli sono visibili nella sezione sommario. Ogni capitolo è composto da paragrafi. Gli utenti watterpad possono rilasciare commenti sui singoli paragrafi dei libri.

Nella sezione "ti piacerà anche" viene associati al libro corrente una lista di libri correlati. Due libri sono considerati correlati se hanno almeno 3 tag in comune e sono stati entrambi votati da almeno 10 utenti.

---

La pagina principale di un utente è costituita dalla sua immagine di profilo, il suo nickname, il Nome e Cognome e le informazioni di base.

Troviamo anche un contatore che tiene traccia rispettivamente dei libri in elenco di lettura e delle opere prodotte dall'utente e degli utenti che lo seguono (followers). In un'area apposita ci sono le informazioni di base sull'utente come la data della sua iscrizione e i vari link ai suoi profili social.

Nella sezione LIBRO troviamo l'immagine di copertina del libro con il titolo, il numero che identifica il libro all'interno di una serie (es [5/6] indica il quinto libro di una serie di sei libri), il numero delle letture (abbreviazione usata 4k = 4000 e così via), il numero dei voti ricevuti e il numero dei capitoli di cui è composto. Subito sotto troviamo un link al profilo dell'autore con immagine di profilo, username e lo stato di

avanzamento del libro con relative date di aggiornamento (es. Dic 10,2017),con relativi link ai social dell'autore.

Troviamo a seguire un collegamento tramite il quale è possibile aggiungere il libro alla lista lettura,troviamo inoltre una breve ma concisa descrizione del libro fornita dall'autore.

Ci sono dei TAG che identificano il libro, la sezione con i commenti ed il sommario per degli approfondimenti.

Come ultimo vi è una sezione in cui a seconda dei TAG del libro troveremo i libri correlati consigliati all'utente con le varie descrizioni sul titolo, il numero di lettura ed i pareri positivi ottenuti dallo stesso.

## **Requisiti Ristrutturati in gruppi di frasi omogenee**

**informazioni generali.** Si vuole realizzare una base di dati per la gestione di una piattaforma in cui gli utenti possono leggere e scrivere libri oltre che interagire fra loro.

**Informazioni su Libro.** Un'opera prodotta da un utente che può essere letta, salvata nella lista lettura e commentata.

**Informazioni sull'utente.** Per gli utenti rappresentiamo nome, cognome,nickname, immagine e link al proprio profilo, oltre al numero di libri apprezzati .

**Informazioni sui follower.** Per i follower indichiamo il numero degli stessi.

**informazioni sullo scrittore.** Per lo scrittore viene indicato il nome quindi l'autore del libro, l'ultimo aggiornamento del relativo libro ed un link in cui è possibile visitare la pagina personale dello scrittore.

**informazioni sul lettore.** Un utente che legge un determinato libro. Può valutarlo mettendo un "mi piace" o commentarlo.

**Informazioni sulla lista lettura.** Questa sezione indica i libri messi in coda ossia quelli dedicati ad una successiva lettura.

**informazioni su categorie.** Questa rappresenta un insieme di libri accomunati dallo stesso genere, infatti ogni categoria contiene libri simili tra loro, ogni categoria associa a sé anche un insieme di scrittori ossia quelli che hanno scritto almeno due

libri nella determinata categoria. Vi sono anche categorie come PRIMO PIANO, IN SALITA o NUOVO.

**informazioni su capitolo.** I capitoli servono a suddividere il libro in sezioni più piccole, il sommario terrà traccia di ognuna di esse. Ogni capitolo è composto da paragrafi.

**informazioni su sommario.** Il sommario serve a tenere traccia di ogni capitolo tramite il titolo di ognuno di essi.

**informazioni su paragrafo.** Ogni paragrafo rappresenta un sottoinsieme di un capitolo. Un utente può rilasciare un commento su un paragrafo.

**informazioni su commento.** Un utente può rilasciare, in una sezione appositamente dedicata, un'opinione personale a proposito di un libro, questo è il commento.

**informazioni su TAG.** Ciascun libro può essere catalogato da un TAG. Se due libri hanno almeno 3 TAG in comune ed entrambi hanno ricevuto un numero di voti uguale almeno a 10 essi sono considerati correlati.

**informazioni su libri correlati.** Correlati è una proprietà che accomuna due o più libri se questi hanno almeno 3 TAG uguali, oltre ad avere altre caratteristiche richieste. Stesso discorso vale anche per gli scrittori ossia se due autori hanno scritto almeno 2 libri nella stessa categoria. Vi è infine la sezione "ti piacerà anche " in cui al libro corrente viene associato una lista di libri ritenuti simili.

**informazioni su voto.** Ciascun utente può esprimere il suo giudizio riguardo un libro tramite un "mi piace", informazione che viene mostrata nel profilo dell'utente.

## Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Libro	Il libro scritto da un utente	Opera	Utente, Capitolo, Scrittore, Lettore, Sommario, Paragrafo, Commento, Categoria
Utente	Generalizzazione	Scrittore/ Lettore	Scrittore, Lettore, Commento
Follower	Utente che ne segue un altro		Utente, Scrittore, Lettore
Scrittore	Utente che scrive un libro	Utente, Autore	Libro, Utente, Lettore, Conversazione, Commento
Lettore	Utente che legge un libro	Utente	Scrittore, Libro, Utente
Lista Lettura	Libri che un utente salva per successiva lettura		Libro, Lettore, Scrittore
Categorie	Gruppi tematici di libri	Genere narrativo	Libro, Lettore, Scrittore
Capitolo	Sezione di un libro		Libro, Lettore, Scrittore, Commento
Sommario	Indice dei capitoli	Indice	Libro, Lettore, Scrittore
Paragrafo	Sotto sezione di un libro		Libro, Lettore, Scrittore, Commento
Commento	Opinione espressa da utente, recensione	Recensione	Libro, Lettore, Scrittore
Tag	Etichetta per catalogare libro	Etichetta	Libro, Categoria, Scrittore, Lettore, Lista Lettura

Libri Correlati	Libri che hanno almeno 3 tag in comune e votati almeno da 10 utenti		Libro, Lettore, Scrittore, Commento, Tag, Voto
Voto	Valutazione positiva espressa da utente riguardo un libro	Valutazione, Parere, Mi Piace, Like	Libro, Lettore, Scrittore, Commento, Tag, Libro Correlato
Volume	Posizione del libro all'interno di una collana		Libro, Scrittore

[illegible]

**RA08:** Il genere narrativo di un determinato libro indica l'argomento e quindi il genere letterario che tale libro descrive, esempio: poesia, romanzo, giallo ecc.

**RA09:** Ad ogni categoria narrativa vengono associati un insieme di scrittori correlati composto da tutti gli scrittori che hanno scritto almeno 2 libri nella categoria narrativa.

## PROGETTAZIONE LOGICA

### Analisi dei volumi

Per il concetto di *Utente* abbiamo ricercato il numero di utenti iscritti a Wattpad che nell'anno 2018 è risultato essere di circa 65 M.

Per il concetto di *Libro* abbiamo ricercato il numero di storie caricate su Wattpad che nell'anno 2017 è risultato essere di circa 400 M.

Per il concetto di *Categoria* sappiamo che il loro numero è 3.

Per il concetto di *Sommario* sappiamo che ogni libro ne ha uno, quindi avremo 400 M di sommari.

Per il concetto di *Capitolo*, sappiamo che in media un sommario ha 10 capitoli, quindi in totale avremo  $400 * 10 = 4000$  M.

Per il concetto di *Paragrafo* sappiamo che in media un Capitolo è formato da 10 Paragrafi, quindi avremo  $4000 * 10 = 40000$  M.

Per il concetto di *Commento* sappiamo che in media viene rilasciato un commento ogni 2 Paragrafi, quindi il loro numero è di 20000 M.

Per la relazione *Lista* ipotizziamo che un utente abbia la propria lista personale, quindi il volume è  $65M * 5 = 325$  M.

Per la relazione *Scrive*, sappiamo che ci sono 65 M di utenti e 400 M di storie, quindi il volume di Scrive è 400 M.

Per la relazione *Legge*, sappiamo che tutti i lettori leggono almeno un libro, quindi stimiamo il volume in 100M.

Per la relazione *Segue* sappiamo che un Utente segue in media 32 altri utenti, quindi il volume è  $65M * 32 = 2080$  M.



Per la relazione “È” riferita al fatto che esistono 3 categorie e 400 M di libri, di questi solo 10 M sono inseriti in queste categorie, quindi il volume è 10 M.

Per la relazione *Ha* riferita alla relazione (*ha sommario*) indica che un libro ha un solo sommario, quindi abbiamo 400 M.

Per la relazione *Composto Paragrafi* sappiamo che un capitolo ha 10 Paragrafi quindi avremo 40000 M.

Per la relazione *Fa* riferito al fatto che un utente può commentare su un determinato paragrafo, inoltre sappiamo che un paragrafo su due riceve un commento, quindi abbiamo 20000M di commenti.

Per la relazione *Correlato* stimiamo ci siano 10K di scrittori correlati.

## TAVOLA DEI VOLUMI

Concetto	Tipo	Volume
Utente	E	65 M
Libro	E	400 M
Lista	R	325 M
Categoria	E	3
Di	R	400 M
Capitolo	E	4000 M
Paragrafo	E	40000 M
Commento	E	20000 M
Scrive	R	400 M
Legge	R	100 M
Segue	R	2080 M
È	R	10 M

Ha	R	400 M
Composto Paragrafi	R	40000 M
Fa	R	20000 M
Correlato	R	10K

### Elenco delle Operazioni Principali

#### TAVOLA DELLE OPERAZIONI

Operazione	Descrizione	Tipo	Frequenza
Op1	Creare Utente	I	200K
Op2	Cancella Utente	I	1K
Op3	Creare Libro	I	10K
Op4	Commentare	I	1M
Op5	Aggiungere libro a Categoria	I	5K
Op6	Votare Libro	I	5M
Op7	Aggiungere libro in Lista	I	100M
Op8	Seguire Utente	I	325M
Op9	Aggiungere tag	I	70K

**Op1-2:** Creare e/o cancellare un Utente

Frequenza: Si stima che il numero di utenti nuovi al mese sia di 200k. Mentre quello di utenti cancellati è di 1000.

**Op3:** Caricare un libro

Frequenza: Si stima che ogni mese vengono caricati circa 10000 nuovi libri.

**Op4:** Commentare un paragrafo di un libro

Frequenza: Sappiamo che un paragrafo su due viene commentato, considerando che al mese abbiamo 10000 nuovi libri, ciascuno con circa 100 paragrafi, stimiamo il numero di operazioni mensili in 1 M.

**Op5:** Aggiungere ad una delle tre categorie un libro

Frequenza: Si stima che un libro viene aggiunto ad una delle categorie mediamente 5000 volte al mese.

**Op6:** Votare un libro

Frequenza: Si stima che un libro in media riceve 500 voti al mese, per 10000 libri avremo 5 M di voti.

**Op7:** Aggiungere un libro in lista lettura

Frequenza: Si stima che un libro viene aggiunto in lista lettura in media 1 volta su 4, quindi approssimativamente avremo 100M di libri nuovi in lista mensilmente.

**Op8:** Seguire un utente (Utente 2 segue Utente 1)

Frequenza: Si stima che ogni utente ne segue 5 nuovi al mese, quindi si stimano  $5 \times 65M = 325$  M di operazioni al mese.

**CON RIDONDANZA:** Quando un utente inizia a seguirne un altro abbiamo due accessi all'entità Utente (per l'utente 1 e per l'utente 2), uno all'associazione "Segui", ed infine l'aggiornamento al contatore "Numero Follower". Quindi avendo  $65M \times 5$  nuovi seguiti al mese sono 325 M. Aggiungendo 200k nuovi iscritti  $\times 5$  seguiti sono 1M di operazioni. In tutto 326 M. Questo significa che avremo  $326M \times 5$  (accedi due volte a utente, aggiorna relazione, aggiorna contatore follower, aggiorna contatore seguiti) = 1630 M operazioni

<b>Tavola degli accessi con ridondanza</b>
--

Concetto	Costrutto	Tipo	Accessi
Utente	E	L	2
Segue	R	S	1
N° Follower Utente 1	A	S	1

N° Seguiti Utente 2	A	S	1
---------------------	---	---	---

**SENZA RIDONDANZA:** Quando un utente inizia a seguirne un altro abbiamo invece due accessi all'entità Utente (utente 1 e utente 2 ) ed uno all'associazione "Segui". Quindi avremo  $326 M * 4$  (accedi due volte a utente, aggiorna relazione, aggiorna follower) = 1304 M.

<b>Tavola degli accessi senza ridondanza</b>
--

Concetto	Costrutto	Tipo	Accessi
Utente	E	L	2
Segui	R	S	1
N° Follower	A	S	1

**Op9:** Aggiungere un tag ad un libro

Frequenza: Il tag viene aggiunto con la creazione del libro e in media ogni libro ne ha 7, quindi avremmo  $7 * 10000 = 70000$  operazioni al mese.

# Ristrutturazione dello schema E-R

## Analisi delle ridondanze

### Utente

#### Presenza di ridondanza:

**Spazio:** assumiamo di usare un intero (4 byte) per rappresentare il numero di followers e un'altro per il numero dei seguiti.

#### Tempo:

1. *Operazione 1:* creare una tabella utente richiede lo stesso numero di accessi sia con ridondanza che senza.
2. *Operazione 2:* stesso tempo che senza ridondanza.
3. *Operazione 8:* 5 accessi: uno all'associazione segui e due all'entità utente, un aggiornamento di *N° Followers* e uno del *N° Seguiti*.

#### Assenza di ridondanza:

**Spazio:** 0 byte.

1. *Operazione 1:* creare una tabella utente richiede lo stesso numero di accessi sia con ridondanza che senza.
2. *Operazione 2:* stesso tempo che con ridondanza.
3. *Operazione 8:* accesso solo all'associazione

Si risparmiano 326M di accessi mensili se eliminiamo la ridondanza.

## Scelta degli identificatori principali

### Utente:

Il *Nickname* dell'utente è una stringa che lo identifica univocamente all'interno del sistema. Con una lunghezza di massimo 20 caratteri alfanumerici, riusciamo a coprire ampiamente un enorme bacino di utenti garantendo l'unicità del nickname e spazio per un discreto livello di personalizzazione.

Il Nome dell'utente è un vettore di 30 caratteri mentre per memorizzare il numero di follower usiamo un intero su 4 byte.

### Libro:

La chiave primaria è un ID di tipo intero su 4 byte. Anche se il volume di questi è di 400M rientra nel range di un intero (-2,147,483,648 fino a 2,147,483,647).

In questo modo abbandoniamo la precedente chiave primaria di libro composta da Autore, Titolo, Volume semplificando la gestione dell'entità e ottimizzando lo schema. Per il Titolo, e il Genere e usiamo due vettori di char, impostiamo rispettivamente un limite di 50 e 50 caratteri occupando 50 byte per il Titolo, 50 byte per il Genere ,per il Volume utilizziamo un intero su 4 byte, per i "Mi Piace" usiamo un smallint occupando 16 bit(2 byte) e infine per Aggiornamento il formato Data g/m/a.

### **Categoria:**

La chiave primaria è un attributo "nome", infatti risulta sensato dato che abbiamo solo 3 categorie.

### **Capitolo:**

Abbiamo una chiave primaria rappresentata dall'ID, utilizziamo perciò un int di 4 byte . Stimiamo il numero dei capitoli medi in 10, considerando  $400M \text{ libri} * 10 = 4000 \text{ M di capitoli}$ . Ciò va oltre i limiti dell'intero su 4 byte. Si utilizza quindi un long long int che ha un range di  $[-9,223,372,036,854,775,807, +9,223,372,036,854,775,807]$ .

### **Paragrafo:**

Anche in questo caso abbiamo una chiave primaria rappresentata dall'ID ed anche qui si utilizza un int di 4 byte. Stimiamo il numero dei paragrafi medi per capitolo in 10 unità, considerando  $400M \text{ libri} * 10 * 10 = 40000 \text{ M di capitoli}$ . Ciò va oltre i limiti dell'intero su 4 byte . Quindi usiamo un long long int che ha un range di  $[-9,223,372,036,854,775,807, +9,223,372,036,854,775,807]$ .

Per contenere il testo del paragrafo usiamo un vettore di char della lunghezza massima di 3000 caratteri, occupando 3K byte.

### **Commento:**

Per l'entità commento infine usiamo un long long int per il suo ID. Stimiamo il numero dei commenti medi per paragrafo in 2.

Considerando  $400M \text{ libri} * 10 * 10 * 2 = 80000 \text{ M di capitoli}$ . Per il campo di testo impostiamo un massimo di 200 caratteri memorizzabili come un array di char, quindi 200 byte.

### **Tag:**

Per l'entità *Tag* utilizziamo un vettore di char per il suo nome, per contenerlo impostiamo un massimo di 20 caratteri, occupando 20 byte.

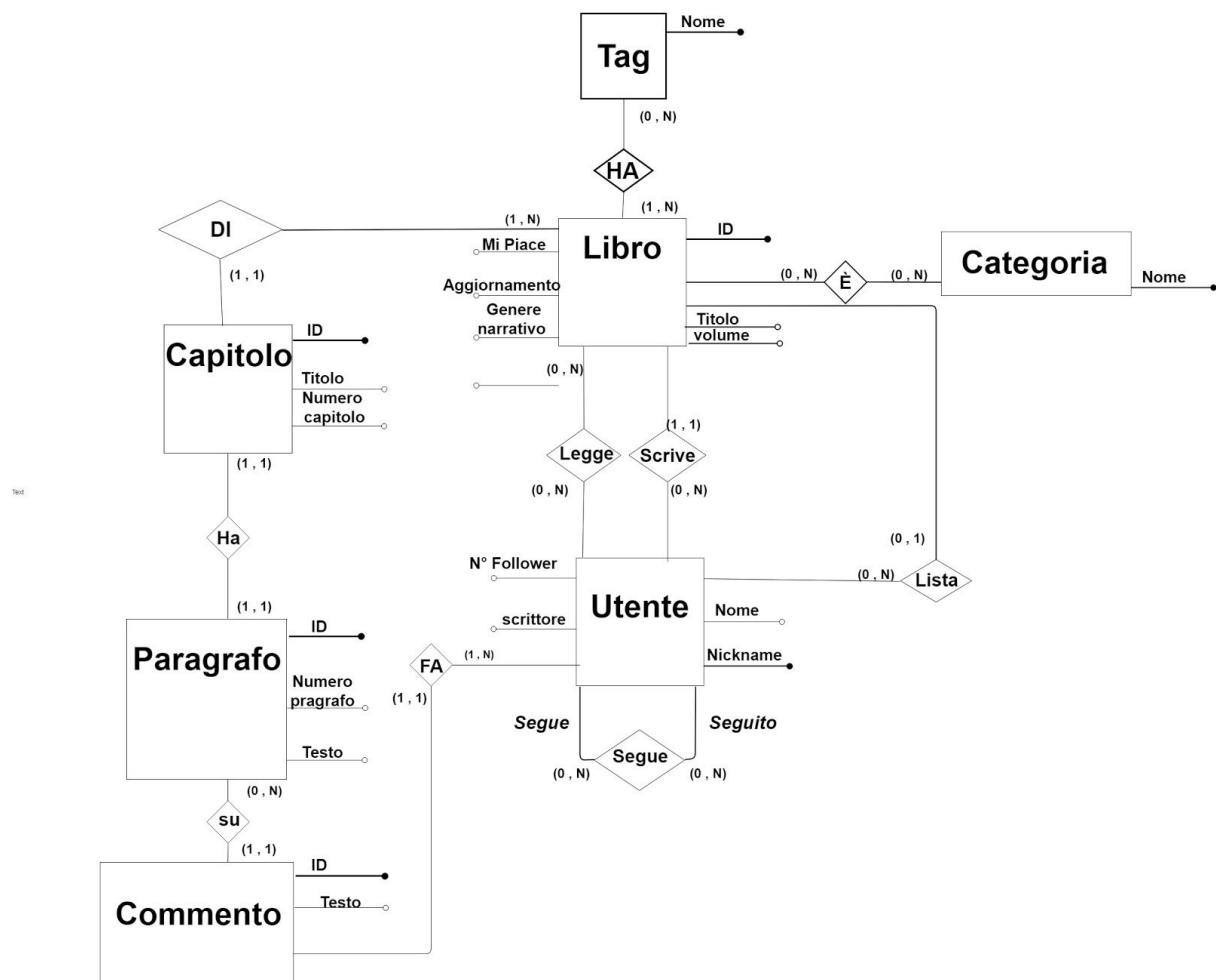
## **Eliminazione delle generalizzazioni**

### **Utente**

La generalizzazione che coinvolge l'entità *utente* e *scrittore* rappresenta l'insieme completo degli utenti del sito. Le strategie a disposizione per eliminare le

generalizzazione sono diverse, ma considerando la natura delle operazioni che dobbiamo effettuare e il volume dei dati, riteniamo che quella più efficiente sia l'accorpamento dell'entità figlia nel padre, ottenendo una sola tabella *Utente*. L'associazione *scrive* che in precedenza riferiva alla singola sottoentità *scrittore* ora collega direttamente *Utente* con *Libro*.

### Schema E-R ristrutturato + Vincoli Aziendali



**RA01:**Un libro deve avere almeno un tag e un titolo.

**RA02:**La relazione lista rappresenta l'insieme di libri destinati ad una futura lettura.

**RA03:** Il testo di un Paragrafo non può essere vuoto.

**RA04:** Perché due libri siano correlati devono aver ricevuto almeno 10 voti e avere almeno 3 tag in comune.

**RA05:** Due scrittori sono correlati se hanno scritto almeno 2 libri nella stessa categoria narrativa.

**RA06:** Categoria può assumere solo 3 valori: “In Primo Piano”, “Nuovi” e “In Salita”.

**RA07:** La data di aggiornamento di un libro deve indicare l’ultima data in cui è stata effettuata una modifica al libro, altrimenti la data in cui il libro è stato rilasciato. Non può essere NULL.

**RA8:** Il nome di un Tag non può essere NULL, tutti i TAG devono avere nome diverso.

**RA9:** Capitolo ha un titolo non NULL, numero capitolo non NULL .

**RA10:** Numero paragrafo non NULL , testo non NULL.

**RA11:** Commento ha un testo non vuoto altrimenti non esiste nemmeno, e si riferisce ad un Paragrafo non NULL.

**RA12:** Il Nickname utente deve essere unico.

### **Schema relazionale + Vincoli di Integrità referenziale**

**UTENTE**(ID, Legge, Scrive, n\_Follower, Segue, Lista, Nome, Nickname)

UTENTE.Legge -> LEGGE.Utente

UTENTE.Scrive -> SCRIVE.Utente

UTENTE.Segue -> SEGUE.Utente1

UTENTE.Lista -> LISTA.ID

**LEGGE**(Utente, Libro)

LEGGE.Utente -> UTENTE.ID

LEGGE.Libro -> LIBRO.ID

**SCRIVE**(Utente, Libro)

SCRIVE.Utente -> UTENTE.ID

SCRIVE.Libro -> LIBRO.ID

**SEGUE**(Utente1, Utente2)

SEGUE.Utente1-> UTENTE.ID

SEGUE.Utente2-> UTENTE.ID



*Nota: Utente1 segue Utente2, quindi il primo è follower.*

**LISTA(Utente,Libro)**

LISTA.Utente -> UTENTE.ID

LISTA.Libro -> LIBRO.ID

**LIBRO(ID,Categoria,Tag,Mi\_Piace,Aggiornamento,Genere,Volume,Titolo)**

LIBRO.Categoria- > CATEGORIA.ID

**HA\_TAG(Libro,Tag)**

HA\_TAG.Libro -> LIBRO.ID

HA\_TAG.Tag -> TAG.ID

**TAG(Nome,Libro)**

TAG.Libro -> LIBRO.ID

**CATEGORIA(ID,Libro,Nome)**

CATEGORIA.Libro -> LIBRO.ID

**HA\_CATEGORIA(Libro,Categoria)**

HA\_CATEGORIA.Libro-> LIBRO.ID

HA\_CATEGORIA.Categoria -> CATEGORIA.ID

**SOMMARIO(ID,Libro)**

SOMMARIO.Libro->LIBRO.ID

**HA\_SOMMARIO(Libro,Sommario)**

HA\_SOMMARIO.Libro -> LIBRO.ID

HA\_SOMMARIO.Sommario -> SOMMARIO.ID

**CAPITOLO(ID,Sommario,Titolo,Pag\_Iniziale,N°\_Capitolo)**

CAPITOLO.Sommario->SOMMARIO.ID

**HA\_CAPITOLO(Sommario,Capitolo)**

HA\_CAPITOLO.Sommario -> SOMMARIO.ID

HA\_CAPITOLO.Capitolo -> CAPITOLO.ID

**PARAGRAFO(ID,Capitolo,Titolo,N°\_Pagina, Testo)**

PARAGRAFO.Capitolo -> CAPITOLO.ID

**HA\_PARAGRAFO(Capitolo,Paragrafo)**

HA\_PARAGRAFO.Capitolo -> CAPITOLO.ID

HA\_PARAGRAFO.Paragrafo -> PARAGRAFO.ID

**COMMENTO**(ID,Paragrafo,Nome\_Utente,Testo)

COMMENTO.Paragrafo -> PARAGRAFO.ID

COMMENTO.Nome\_Utente -> UTENTE.ID

**SU\_COMMENTO**(Commento,Paragrafo)

SU\_COMMENTO.Paragrafo -> PARAGRAFO.ID

SU\_COMMENTO.Commento -> COMMENTO.ID

**SCRIVE\_COMMENTO**(Commento, Utente)

SCRIVE\_COMMENTO.Commento -> COMMENTO.ID

SCRIVE\_COMMENTO.Utente -> UTENTE.ID

## DDL

```
CREATE TABLE LIBRO (  
    ID SERIAL,  
    titolo VARCHAR(30) NOT NULL,  
    volume integer,  
    like integer,  
    aggiornamento timestamp,  
    genere VARCHAR(20),  
    PRIMARY KEY (ID)  
);
```

```
CREATE TABLE UTENTE (  
    ID SERIAL,  
    N_FOLLOWER INTEGER NOT NULL,  
    NOME VARCHAR(30) NOT NULL,  
    Nickname VARCHAR(20) NOT NULL,  
    PRIMARY KEY (ID)  
);
```

```
CREATE TABLE TAG(  
    Nome VARCHAR(30) NOT NULL UNIQUE,  
    Libro integer,
```

FOREIGN KEY (Libro) REFERENCES LIBRO(ID) ON DELETE CASCADE  
ON UPDATE CASCADE

);

CREATE TABLE PARAGRAFO(  
ID SERIAL UNIQUE,  
Testo varchar(3000)

);

CREATE TABLE COMMENTO(  
ID SERIAL UNIQUE,  
Testo VARCHAR(300) NOT NULL

);

CREATE TABLE CAPITOLO(  
ID SERIAL UNIQUE,  
N\_Pagina integer,  
Titolo VARCHAR(30) NOT NULL

);

CREATE TABLE CATEGORIA(  
ID SERIAL UNIQUE,  
Nome VARCHAR(30) NOT NULL,

);

/\* RELAZIONI \*/

CREATE TABLE LEGGE (  
Utente Integer,  
Libro Integer,  
FOREIGN KEY (Utente) REFERENCES UTENTE (ID) ON DELETE  
CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (Libro) REFERENCES LIBRO (ID) ON DELETE CASCADE  
ON UPDATE CASCADE

);

```
CREATE TABLE SCRIVE (  
    Utente Integer,  
    Libro Integer,  
    FOREIGN KEY (Utente) REFERENCES UTENTE (ID) ON DELETE  
    CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (Libro) REFERENCES LIBRO (ID) ON DELETE CASCADE  
    ON UPDATE CASCADE
```

);

```
CREATE TABLE SEGUE (  
    Utente1 Integer,  
    Utente2 Integer,  
    FOREIGN KEY (Utente1) REFERENCES UTENTE (ID) ON DELETE  
    CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (Utente2) REFERENCES UTENTE (ID) ON DELETE  
    CASCADE ON UPDATE CASCADE
```

);

```
CREATE TABLE LISTA (  
    Utente Integer,  
    Libro Integer,  
    FOREIGN KEY (Utente) REFERENCES UTENTE (ID) ON DELETE  
    CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (Libro) REFERENCES LIBRO (ID) ON DELETE CASCADE  
    ON UPDATE CASCADE
```

);

```
CREATE TABLE IN_CATEGORIA(  
    Libro integer,  
    Categoria integer,  
    FOREIGN KEY (Libro) REFERENCES LIBRO(ID) ON DELETE CASCADE  
    ON UPDATE CASCADE,  
    FOREIGN KEY (Categoria) REFERENCES CATEGORIA(ID) ON DELETE  
    CASCADE ON UPDATE CASCADE
```

);

```
CREATE TABLE HA_TAG(  
    Libro Integer,  
    Tag character varying(30),  
    FOREIGN KEY (Libro) REFERENCES LIBRO(id) ON DELETE CASCADE  
    ON UPDATE CASCADE,  
    FOREIGN KEY (Tag) REFERENCES TAG(Nome) ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

```
CREATE TABLE SOMMARIO(  
    Libro integer,  
    Capitolo integer,  
    FOREIGN KEY (Libro) REFERENCES LIBRO (ID)ON DELETE CASCADE  
    ON UPDATE CASCADE,  
    FOREIGN KEY (Capitolo) REFERENCES Capitolo (ID) ON DELETE  
    CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE SU_PARAFRAGO(  
    Commento integer,  
    Paragrafo integer,  
    FOREIGN KEY (Commento) REFERENCES COMMENTO(ID) ON DELETE  
    CASCADE ON UPDATE CASCADE ,  
    FOREIGN KEY (Paragrafo) REFERENCES PARAGRAFO(ID) ON DELETE  
    CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE FA_COMMENTO(  
    Commento integer,  
    Utente integer,  
    FOREIGN KEY (Commento) REFERENCES Commento(ID) ON DELETE  
    CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (Utente) REFERENCES UTENTE(ID) ON DELETE  
    CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE HA_PARAGRAFO(  
    Capitolo Integer,  
    Paragrafo Integer,
```

FOREIGN KEY (Capitolo) REFERENCES CAPITOLO (ID) ON DELETE  
CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (Paragrafo) REFERENCES PARAGRAFO (ID) ON DELETE  
CASCADE ON UPDATE CASCADE

);

## DML

INSERT INTO Categoria VALUES (1,'Nuovo');  
INSERT INTO Categoria VALUES (2,'In Salita');  
INSERT INTO Categoria VALUES (3,'Primo Piano');

INSERT INTO utente (nome, nickname) VALUES  
( 'Silvestro' , 'Silvestro96'),  
( 'Aldo', 'Aldo77'),  
( 'Mikelian', 'Miky69'),  
( 'Antonino', 'Antonino33');

INSERT INTO libro (Titolo, Genere, Volume, like) VALUES  
( 'Il Signore degli Anelli', 'Fantasy', '1', '1000'),  
( 'Alice nel Paese delle Meraviglie', 'Per Ragazzi', '1', '100');

INSERT INTO libro (Titolo, Autore, Genere, Volume) VALUES  
( 'Il Signore degli Anelli', 'Fantasy', '1'),  
( 'Le 1000 e una Notte', 'Per Ragazzi', '1');

INSERT INTO capitolo (pag\_iniziale, titolo)  
VALUES (10, 'Gli Hobbit');

INSERT INTO Sommario VALUES (2,1);

INSERT INTO Paragrafo (Testo) VALUES  
( 'Nove saranno i membri della Compagnia dell Anello,  
e i Nove Viandanti si opporranno ai Nove Cavalieri che sono malvagi.  
Con te e con il tuo fido servo verrà Gandalf; questo sarà infatti il suo grande  
incarico,

e forse la fine dei suoi travagli.

Gli altri rappresenteranno i rimanenti popoli liberi della Terra: Elfi, Nani e Uomini.');

```
INSERT INTO COMMENTO (testo)VALUES
('Eccellente!!'),
('Meraviglioso'),
('Che bella storia!!');
```

```
INSERT INTO FA_COMMENTO VALUES
(1,1),
(1,3),
(2,2);
```

```
INSERT INTO IN_CATEGORIA VALUES
(2,1),
(3,2);
```

*Qualche operazione di modifica:*

Un Utente vuole cambiare il suo Nickname perchè quello che ha non gli piace più o aveva sbagliato a digitarlo.

```
UPDATE UTENTE SET NICKNAME='SILVIO96' WHERE NOME='Silvestro';
```

*Prima della modifica:*

	id [PK] integer	nome character varying (30)	nickname character varying (20)
1	1	Silvestro	Silvestro96
2	2	Antonino	Antonino33
3	3	Aldo	Aldo77
4	4	Mikelian	Miky69

*Dopo la modifica:*

	id [PK] integer	nome character varying (30)	nickname character varying (20)
1	1	Silvestro	SILVIO96
2	2	Antonino	Antonino33
3	3	Aldo	Aldo77
4	4	Mikelian	Miky69

Notare come il Nickname del primo utente sia cambiato.