



Inserire qui il titolo

Inserire il nome
Inserire la matricola
Inserire la data

Indice

1	Requisiti identificati	4
2	Progettazione concettuale	6
2.1	Class Diagram	6
2.2	Analisi della ristrutturazione del Class Diagram	7
2.2.1	Analisi delle ridondanze	7
2.2.2	Analisi degli identificativi	7
2.2.3	Rimozione degli attributi multivalore	7
2.2.4	Rimozione degli attributi composti	7
2.2.5	Partizione/Accorpamento delle associazioni	8
2.2.6	Rimozione delle gerarchie	8
2.3	Class Diagram ristrutturato	9
2.4	Dizionario delle classi	10
2.5	Dizionario delle associazioni	12
3	Schema logico	13
4	Schema Fisico	15
4.1	Creazione Tabelle	15
4.1.1	Tabella Articoli	15
4.1.2	Tabella Autore	15
4.1.3	Tabella AutoreArticolo	16
4.1.4	Tabella Riviste	16
4.1.5	Tabella ArticoliInRiviste	16
4.1.6	Tabella Evento	17
4.1.7	Conferenza	17
4.1.8	Tabella Libri	17
4.1.9	Tabella AutoreLibro	18
4.1.10	Tabella Presentazione	18
4.1.11	Tabella Serie	18
4.1.12	Tabella LibriInSerie	19
4.1.13	Tabella Negozio	19
4.1.14	Tabella Stock	19
4.1.15	Tabella Utente	19

4.1.16	Tabella Richiesta	20
4.2	Creazione Funzioni e Procedure	20

Capitolo 1

Requisiti identificati

Si vuole sviluppare un sistema informativo di gestione di una biblioteca digitale contenente **Libri** e **Articoli scientifici**.

I libri possono essere **Didattici** o **Romanzi**.

In particolare, questi ultimi possono essere parte di **Collane**, raggruppate per caratteristiche comuni, e appartenere ad una **Serie** se hanno uno o più seguiti, gli articoli possono essere parte di una **Rivista** oppure essere presentati durante una **Conferenza**.

Il sistema dovrà inoltre permettere ad un **Utente** la ricerca di un libro e recuperare la lista di **Negozi** in cui sia possibile acquistare quest'ultimo. L'utente potrà inoltre ricercare una serie (o collana) di libri e un negozio in cui quest'ultima potrà essere acquistata nel caso in cui al momento della ricerca non ci fosse alcun negozio idoneo, l'utente potrà inoltrare una **Richiesta** di notifica nel momento in cui uno dei negozi avrà tutti i libri appartenenti alla serie.

In particolare sono state identificate le seguenti entità:

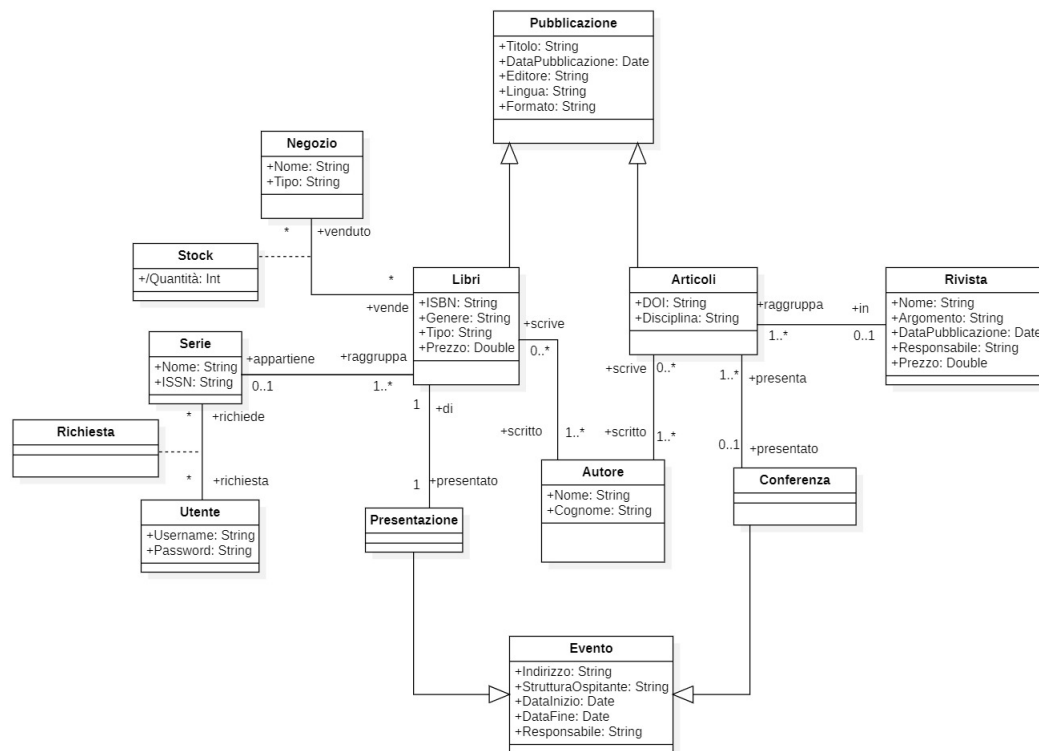
1. **Pubblicazione**: Generalizzazione di un libro o un articolo scientifico
2. **Libro**: Specializzazione di una Pubblicazione
3. **Articolo scientifico**: Specializzazione di una Pubblicazione
4. **Rivista**: Entità che identifica un insieme di articoli
5. **Evento**: Generalizzazione di una Conferenza o di una Presentazione
6. **Conferenza**: Specializzazione di un Evento
7. **Presentazione**: Specializzazione di un Evento
8. **Autore**: Entità che identifica l'autore di un Libro o di un Articolo
9. **Negozio**: Entità che identifica un Negozio
10. **Serie**: Entità che identifica un insieme di libri con caratteristiche simili

11. **Richiesta:** Entità che identifica la richiesta di disponibilità di una serie da parte di un utente

Capitolo 2

Progettazione concettuale

2.1 Class Diagram



2.2 Analisi della ristrutturazione del Class Diagram

In questa fase verranno effettuate delle modifiche al Class Diagram in modo tale da renderlo più adatto a una traduzione al modello logico. (magari scriviamo meglio sta parte)

2.2.1 Analisi delle ridondanze

Nel Diagramma Concettuale non ci sono ridondanze tali da essere eliminate.

2.2.2 Analisi degli identificativi

In questa fase andremo a scegliere un attributo per identificare univocamente le varie entità presenti nello schema precedente, in particolare:

1. L'entità **Libro** presenta l'attributo ISBN che rappresenta una possibile chiave primaria, tuttavia è stato scelto di aggiungere un attributo *ID_Libro* in modo tale da aumentare la velocità di accesso agli indici e garantire l'immutabilità della base di dati.
2. Per **Articolo scientifico** la situazione è analoga, è stato quindi aggiunto un attributo *ID_Articolo*.
3. Nel caso dell'entità **Rivista**, la quale presenta un attributo ISSN che è chiave candidata, di inserire un ulteriore attributo *ID_Rivista*.
4. Sarebbe possibile identificare un **Evento** tramite un insieme piuttosto ampio di attributi, è stato quindi aggiunto un attributo *ID_Evento*.
5. Per lo stesso motivo di Evento, è stato aggiunto alla tabella **Autore** un attributo *ID_Autore*.
6. Dato che l'entità **Negozio** non presenta alcuna chiave candidata, è stato aggiunto l'attributo *ID_Negozio*.
7. È stato deciso di aggiungere a **Serie** un attributo *ID_Serie* per ridurre il volume degli indici associati.

2.2.3 Rimozione degli attributi multivalore

Non sono presenti attributi multivalore.

2.2.4 Rimozione degli attributi composti

Non sono presenti attributi composti.

2.2.5 Partizione/Accorpamento delle associazioni

L'unica associazione 1:1 presente in questo Class Diagram è quella tra un **Libro** e una **Presentazione**, nonostante ciò, è stato deciso di mantenere la classe associativa tra le classi perché abbiamo ritenuto più idoneo avere due classi diverse per associare due tipi di pubblicazione diversi a un Evento.

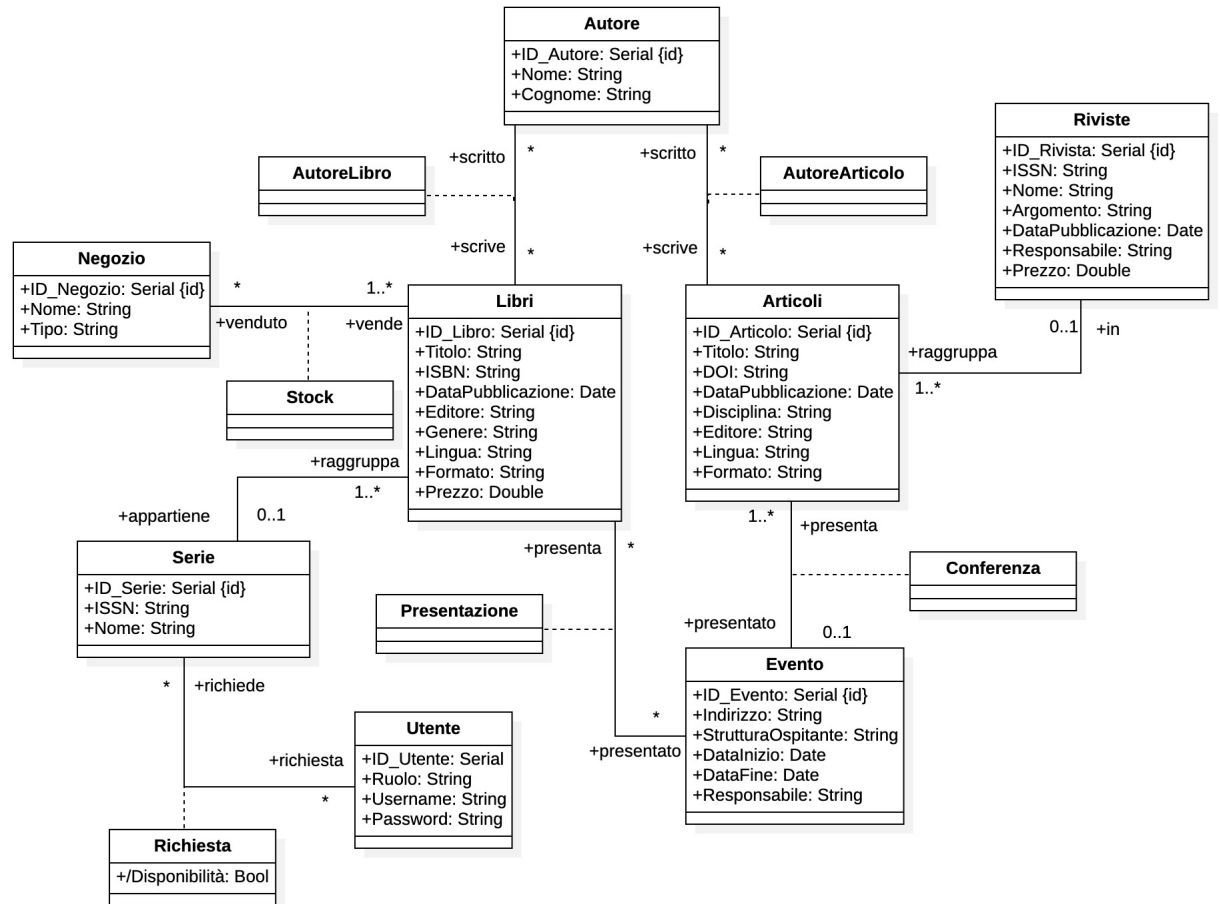
2.2.6 Rimozione delle gerarchie

In questo diagramma sono presenti 2 generalizzazioni e 4 relative specializzazioni. In particolare:

Per quanto riguarda la generalizzazione **Pubblicazione**, si è scelto di accorpare l'entità padre nelle entità figlie, ottenendo come risultato:

- Una entità **Libro** aventi tutti gli attributi di **Pubblicazione** più gli attributi della precedente entità **Libro**.
- Analogamente, l'entità **Articolo** avrà come attributi, quelli di **Pubblicazione** uniti agli attributi di **Articolo**.

2.3 Class Diagram ristrutturato



2.4 Dizionario delle classi

Classe	Attributi	Descrizione
Articolo	ID_Articolo Titolo Doi DataPubblicazione Disciplina Editore Lingua Formato	Classe che conserva le informazioni relative agli articoli presenti nel sistema
Autore	ID_Autore Nome Cognome	Classe che conserva le informazioni relative agli autori presenti nel sistema
Rivista	ID_Rivista ISSN Nome Argomento DataPubblicazione Responsabile Prezzo	Classe che conserva le informazioni relative alle riviste scientifiche presenti nel sistema
AutoreArticolo		Classe che conserva le informazioni sugli autori degli articoli scientifici
ArticoloInRivista		Classe che mette in relazione Articoli e Riviste a cui i primi appartengono
Evento	ID_Evento Indirizzo StrutturaOspitante DataInizio DataFine Responsabile	Classe che conserva le informazioni sugli eventi relativi a presentazioni di libri oppure a conferenze scientifiche
Conferenza		Classe che mette in relazione eventi e articoli scientifici

Classe	Attributi	Descrizione
Libro	ID.Libro Titolo ISBN DataPubblicazione Editore Genere Lingua Formato Prezzo	Classe che conserva le principali informazioni sui libri presenti nel sistema
AutoreLibro		Classe che mette in relazione Autori e Libri
Presentazione		Classe che conserva le informazioni relative a presentazioni di libri
Serie	ID.Serie ISSN Nome	Classe che conserva le informazioni relative alle Serie di libri
Negozio	ID.Negozio Nome Tipo	Classe che conserva le informazioni relative ai Negozi presenti nel sistema
Stock	Quantità	Classe che conserva le informazioni relative ai libri acquistabili da determinati negozi e la relativa quantità
Utente	ID.Utente Username Password	Classe che conserva le informazioni relative agli Utenti registrati sul sistema
Richiesta		Classe che gestisce le richieste fatte dagli utenti relativamente alle disponibilità di serie

2.5 Dizionario delle associazioni

Associazione	Classi coinvolte	Descrizione
scritto ... scrive	Libri e Autore Articoli e Autore.	Uno o più Libri/Articoli vengono scritti da uno o più Autori. [*,*]
venduto ... vende	Negozio, Stock, Libri.	Uno o più negozi vendono uno o più libri. [*,*].
raggruppa ... in	Riviste e Articoli.	Almeno un articolo è pubblicato in una rivista. Una rivista raggruppa diversi articoli. [1..*, 0..1]
presentato ... presenta	Articoli, Conferenza, Evento. Libri, Presentazione, Evento.	Uno o più articoli possono essere presentati in una conferenza Un libro può essere presentato durante un evento. [0..1, 1..*]
raggruppa ... appartiene	Libri, Serie	Una serie raggruppa diversi libri, un libro può appartenere al più a una serie. [1..*, 0..1]
richiede ... richiesta	Utente, Serie	Un utente può richiedere diverse serie. Una serie può essere richiesta da diversi utenti. [*..*]

Capitolo 3

Schema logico

Articolo: ID_Articolo, Titolo, DOI, DataPubblicazione, Editore, Lingua, Formato

Autore: ID_Autore, Nome, Cognome

Rivista: ID_Rivista, ISSN, Nome, Argomento, DataPubblicazione, Responsabile, Prezzo

AutoreArticolo: ID_Autore, ID_Articolo

$ID_Autore \hookrightarrow Autore(ID_Autore)$

$ID_Articolo \hookrightarrow Articolo(ID_Articolo)$

ArticoloInRivista: ID_Rivista, ID_Articolo

$ID_Rivista \hookrightarrow Rivista(ID_Rivista)$

$ID_Articolo \hookrightarrow Articolo(ID_Articolo)$

Evento: ID_Evento, Indirizzo, StrutturaOspitante, DataInizio, DataFine, Responsabile

Conferenza: ID_Evento, ID_Articolo

$ID_Evento \hookrightarrow Evento(ID_Evento)$

$ID_Articolo \hookrightarrow Articolo(ID_Articolo)$

Libro: ID_Libro, Titolo, ISBN, DataPubblicazione, Editore, Genere, Lingua, Formato, Prezzo

AutoreLibro: ID_Autore, ID_Libro

ID_Autore \hookrightarrow Autore(ID_Autore)

ID_Libro \hookrightarrow Libro(ID_Libro)

Presentazione: ID_Evento, ID_Libro

ID_Evento \hookrightarrow Evento(ID_Evento)

ID_Libro \hookrightarrow Libro(ID_Libro)

Serie: ID_Serie, ISSN, Nome

LibroInSerie: ID_Serie, ID_Libro

ID_Serie \hookrightarrow Serie(ID_Serie)

ID_Libro \hookrightarrow Libro(ID_Libro)

Negozio: ID_Negozio, Nome, Tipo

Stock: ID_Negozio, ID_Libro, Quantità

ID_Negozio \hookrightarrow Negozio(ID_Negozio)

ID_Libro \hookrightarrow Libro(ID_Libro)

Utente: ID_Utente, Username, Password

Richiesta: ID_Utente, ID_Serie

ID_Utente \hookrightarrow Utente(ID_Utente)

ID_Serie \hookrightarrow Serie(ID_Serie)

Capitolo 4

Schema Fisico

In questo ultimo capitolo esamineremo i meccanismi necessari per la traduzione di uno schema logico in uno schema fisico. Andremo a definire le tabelle con i relativi attributi e tipi dei dati, le funzioni, le procedure, i trigger e i vincoli. Con questi elementi, sarà possibile creare un database relazionale con una struttura specifica che soddisfi i requisiti identificati nel Capitolo 1.

4.1 Creazione Tabelle

4.1.1 Tabella Articoli

```
1 | CREATE TABLE b.Articoli
2 | ID_Articolo      SERIAL,
3 | Titolo           VARCHAR(128),
4 | DOI              VARCHAR(128),
5 | DataPubblicazione DATE,
6 | Disciplina       VARCHAR(128),
7 | Editore          VARCHAR(128),
8 | Lingua           VARCHAR(128),
9 | Formato          VARCHAR(128),
10 |
11 | CONSTRAINT PK_Articoli PRIMARY KEY (ID_Articolo),
12 | CONSTRAINT UK_Articolo UNIQUE (DOI);
```

4.1.2 Tabella Autore

```
1 | CREATE TABLE b.Autore
2 | ID_Autore SERIAL,
3 | Nome          VARCHAR(128),
4 | Cognome       VARCHAR(128),
5 |
6 | CONSTRAINT PK_Autore PRIMARY KEY (ID_Autore);
```

4.1.3 Tabella AutoreArticolo

```
1 CREATE TABLE b.AutoreArticolo
2 ID_Autore SERIAL,
3 ID_Articolo SERIAL,
4
5 CONSTRAINT PK_AutoreArticolo PRIMARY KEY (ID_Autore, ID_Articolo),
6 CONSTRAINT FK_AutoreArticolo_Autore FOREIGN KEY (ID_Autore)
7 REFERENCES b.Autore (ID_Autore) ON DELETE CASCADE,
8 CONSTRAINT FK_AutoreArticolo_Articoli FOREIGN KEY (ID_Articolo)
9 REFERENCES b.Articoli (ID_Articolo) ON DELETE CASCADE;
```

4.1.4 Tabella Riviste

```
1 CREATE TABLE b.Riviste
2 ID_Rivista SERIAL,
3 ISSN VARCHAR(128),
4 Nome VARCHAR(128),
5 Argomento VARCHAR(128),
6 DataPubblicazione DATE,
7 Responsabile VARCHAR(128),
8 Prezzo FLOAT,
9
10 CONSTRAINT PK_Riviste PRIMARY KEY (ID_Rivista);
```

4.1.5 Tabella ArticoliInRiviste

```
1 CREATE TABLE b.ArticoliInRiviste
2 ID_Articolo SERIAL,
3 ID_Rivista SERIAL,
4
5 CONSTRAINT PK_ArticoliInRiviste PRIMARY KEY (ID_Articolo,
6 ID_Rivista),
7 CONSTRAINT FK_ArticoliInRiviste_Articolo FOREIGN KEY (ID_Articolo)
8 REFERENCES b.Articoli (ID_Articolo) ON DELETE CASCADE,
9 CONSTRAINT FK_ArticoliInRiviste_Rivista FOREIGN KEY (ID_Rivista)
10 REFERENCES b.Riviste (ID_Rivista) ON DELETE CASCADE;
```


4.1.6 Tabella Evento

```
1  ID_Evento      SERIAL,
2  Nome           VARCHAR(128),
3  Indirizzo      VARCHAR(128),
4  StrutturaOspitante VARCHAR(128),
5  DataInizio     DATE,
6  DataFine       DATE,
7  Responsabile   VARCHAR(128),
8
9  CONSTRAINT PK_Evento PRIMARY KEY (ID_Evento),
10 CONSTRAINT CK_Date CHECK (DataInizio <= DataFine),
11 CONSTRAINT UK_Evento UNIQUE (Indirizzo, StrutturaOspitante,
    DataInizio, DataFine, Responsabile);
```

4.1.7 Conferenza

```
1  CREATE TABLE b.Conferenza
2  ID_Articolo SERIAL,
3  ID_Evento SERIAL,
4
5  CONSTRAINT PK_Conferenza PRIMARY KEY (ID_Articolo, ID_Evento),
6  CONSTRAINT FK_Conferenza_Articolo FOREIGN KEY (ID_Articolo)
    REFERENCES b.Articoli (ID_Articolo) ON DELETE CASCADE,
7  CONSTRAINT FK_Conferenza_Evento FOREIGN KEY (ID_Evento) REFERENCES
    b.Evento (ID_Evento) ON DELETE CASCADE;
```

4.1.8 Tabella Libri

```
1  CREATE TABLE b.Libri
2  ID_Libro       SERIAL,
3  Titolo         VARCHAR(128),
4  ISBN          VARCHAR(128),
5  DataPubblicazione DATE,
6  Editore        VARCHAR(128),
7  Genere         VARCHAR(128),
8  Lingua         VARCHAR(128),
9  Formato        VARCHAR(128),
10 Prezzo        FLOAT,
11
12 CONSTRAINT PK_Libri PRIMARY KEY (ID_Libro),
13 CONSTRAINT UK_Libro UNIQUE (ISBN),
14 CONSTRAINT CK_Libri CHECK (Prezzo > 0),
15 CONSTRAINT CK_Titolo (Titolo IS NOT NULL);
```

4.1.9 Tabella AutoreLibro

```
1 CREATE TABLE b.AutoreLibro
2 ID_Autore SERIAL,
3 ID_Libro SERIAL,
4
5 CONSTRAINT PK_AutoreLibro PRIMARY KEY (ID_Autore, ID_Libro),
6 CONSTRAINT FK_AutoreLibro_Autore FOREIGN KEY (ID_Autore)
7 REFERENCES b.Autore (ID_Autore) ON DELETE CASCADE,
8 CONSTRAINT FK_AutoreLibro_Libro FOREIGN KEY (ID_Libro) REFERENCES
9 b.Libri (ID_Libro) ON DELETE CASCADE;
```

4.1.10 Tabella Presentazione

```
1 CREATE TABLE b.Presentazione
2 ID_Evento SERIAL,
3 ID_Libro SERIAL,
4
5 CONSTRAINT PK_Presentazione PRIMARY KEY (ID_Evento, ID_Libro),
6 CONSTRAINT FK_Presentazione_Evento FOREIGN KEY (ID_Evento)
7 REFERENCES b.Evento (ID_Evento) ON DELETE CASCADE,
8 CONSTRAINT FK_Presentazione_Libro FOREIGN KEY (ID_Libro)
9 REFERENCES b.Libri (ID_Libro) ON DELETE CASCADE;
```

4.1.11 Tabella Serie

```
1 CREATE TABLE b.Serie
2 ID_Serie SERIAL,
3 ISSN VARCHAR(128),
4 Nome VARCHAR(128),
5
6 CONSTRAINT PK_Serie PRIMARY KEY (ID_Serie),
7 CONSTRAINT UK_Serie UNIQUE (ISSN);
```

4.1.12 Tabella LibriInSerie

```
1 | CREATE TABLE b.LibriInSerie
2 | ID_Serie INTEGER,
3 | ID_Libro INTEGER,
4 |
5 | CONSTRAINT PK_LibriInSerie PRIMARY KEY (ID_Serie, ID_Libro),
6 | CONSTRAINT FK_Libri_Serie FOREIGN KEY (ID_Serie) REFERENCES
   | b.Serie (ID_Serie) ON DELETE CASCADE,
7 | CONSTRAINT FK_Serie_Libri FOREIGN KEY (ID_Libro) REFERENCES
   | b.Libri (ID_Libro) ON DELETE CASCADE;
```

4.1.13 Tabella Negozio

```
1 | CREATE TABLE b.Negozio
2 | ID_Negozio SERIAL,
3 | Nome VARCHAR(128),
4 | Tipo VARCHAR(128),
5 |
6 | CONSTRAINT PK_Negozio PRIMARY KEY (ID_Negozio);
```

4.1.14 Tabella Stock

```
1 | CREATE TABLE b.Negozio
2 | ID_Negozio SERIAL,
3 | Nome VARCHAR(128),
4 | Tipo VARCHAR(128),
5 |
6 | CONSTRAINT PK_Negozio PRIMARY KEY (ID_Negozio);
```

4.1.15 Tabella Utente

```
1 | CREATE TABLE b.Utente
2 | ID_Utente SERIAL,
3 | Username VARCHAR(128),
4 | Password VARCHAR(128),
5 | Permessi b.TipoUtente DEFAULT '0',
6 |
7 | CONSTRAINT PK_Utente PRIMARY KEY (ID_Utente),
8 | CONSTRAINT UK_Utente UNIQUE (Username);
```

4.1.16 Tabella Richiesta

```
1 CREATE TABLE b.Richiesta
2   ID_Utente    SERIAL,
3   ID_Serie     SERIAL,
4
5   CONSTRAINT PK_Richiesta PRIMARY KEY (ID_Utente, ID_Serie),
6   CONSTRAINT FK_Richiesta_Utente FOREIGN KEY (ID_Utente) REFERENCES
7     b.Utente (ID_Utente) ON DELETE CASCADE,
   CONSTRAINT FK_Richiesta_Serie FOREIGN KEY (ID_Serie) REFERENCES
     b.Serie (ID_Serie) ON DELETE CASCADE;
```

4.2 Creazione Funzioni e Procedure