



Basi di Dati
Progetto A.A. 2022/2023

SISTEMA INFORMATIVO DI UNA BIBLIOTECA

0280298
Matteo La Gioia

Indice

1.	Descrizione del Minimondo	3
2.	Analisi dei Requisiti.....	4
3.	Progettazione concettuale	7
4.	Progettazione logica.....	10
5.	Progettazione fisica	15

1. Descrizione del Minimondo

- 1 Un circuito di biblioteche mette a disposizione un servizio di prenotazione libri. Si vuole
2 realizzare un sistema informativo di gestione per tali biblioteche, ciascuna caratterizzata da
3 indirizzo, numero di telefono, nome del responsabile ed orario di apertura settimanale.
- 4 Il sistema gestisce libri, le cui copie sono disponibili in un sottoinsieme delle biblioteche, in
5 numero differente. Ciascun libro è associato allo stato “in prestito” / “disponibile”.
- 6 Gli utenti del circuito di biblioteche possono registrarsi fornendo tutte le loro informazioni
7 anagrafiche ed un numero arbitrario di contatti (telefono, cellulare, e-mail), specificando qual
8 è il mezzo di comunicazione preferito con cui vogliono essere contattati.
- 9 All’atto di effettuare un prestito, i bibliotecari possono recuperare la disponibilità delle copie
10 del libro presso la biblioteca in cui essi lavorano. Se il libro è disponibile, il sistema
11 restituisce lo scaffale ed il ripiano in cui può essere prelevato il libro. Se non ne è disponibile
12 alcuna copia, il bibliotecario può verificare in quali altre biblioteche del circuito è presente
13 tale libro ed effettuare una richiesta di trasferimento. In questo caso, il libro viene segnalato
14 come “prestato ad altra biblioteca” e viene tenuta traccia di qual è la biblioteca di partenza e
15 di destinazione. All’atto della consegna della copia, l’utente del servizio può chiedere di
16 trattenere in consultazione in libro per 1, 2, o 3 mesi.
- 17 I bibliotecari hanno la possibilità di generare un report indicante quali libri, prestati dalla
18 biblioteca in cui stanno svolgendo il turno, non sono ancora stati restituiti e le informazioni
19 dell’utente che possiede attualmente la copia del libro. I recapiti consentono al bibliotecario
20 di mettersi in contatto per sollecitare la restituzione.
- 21 Un libro che non è stato prestato nei passati 10 anni viene dismesso dalla biblioteca. Tale
22 operazione viene effettuata dagli amministratori del circuito. Il record ad esso associato non
23 viene eliminato, ma questo non potrà più essere prestato agli utenti. I prestiti in corso restano
24 validi fino alla riconsegna.

2. Analisi dei Requisiti

Identificazione dei termini ambigui e correzioni possibili

Linea	Termine	Nuovo termine	Motivo correzione
5	libro	copia	Molteplice utilizzo nella specifica con utilizzo di sinonimo di copia del libro.
10	libro	copia	Molteplice utilizzo nella specifica con utilizzo di sinonimo di copia del libro.
11	libro	copia	Molteplice utilizzo nella specifica con utilizzo di sinonimo di copia del libro.
13	libro	copia	Molteplice utilizzo nella specifica con utilizzo di sinonimo di copia del libro.
17	libro	copia	Molteplice utilizzo nella specifica con utilizzo di sinonimo di copia del libro.

Specifica disambiguata

Un circuito di biblioteche mette a disposizione un servizio di prenotazione **libri**. Si vuole realizzare un sistema informativo di gestione per tali biblioteche, ciascuna caratterizzata da indirizzo, numero di telefono, nome del responsabile ed orario di apertura settimanale.

Il sistema gestisce **libri**, le cui **copie** sono disponibili in un sottoinsieme delle biblioteche, in numero differente. Ciascuna **copia** è associata allo stato “in prestito” / “disponibile”.

Gli **utenti** del circuito di biblioteche possono registrarsi fornendo tutte le loro **informazioni anagrafiche ed un numero arbitrario di contatti (telefono, cellulare, e-mail)**, specificando qual è il **mezzo di comunicazione preferito** con cui vogliono essere contattati.

All’atto di effettuare un prestito, i bibliotecari possono recuperare la **disponibilità delle copie del libro presso la biblioteca in cui essi lavorano**. Se una **copia è disponibile**, il sistema restituisce lo **scaffale ed il ripiano in cui può essere prelevata una copia**. Se non ne è disponibile alcuna **copia**, il bibliotecario può verificare in quali altre biblioteche del circuito è presente tale **libro** ed effettuare una richiesta di trasferimento. In questo caso, la **copia viene segnalata come “prestato ad altra biblioteca” e viene tenuto traccia di qual è la biblioteca di partenza e di destinazione**. All’atto della consegna della **copia**, **l’utente del servizio può chiedere di trattenere in consultazione in libro per 1, 2, o 3 mesi**.

I bibliotecari hanno la possibilità di generare un report indicante quali **copie, prestate dalla biblioteca in cui stanno svolgendo il turno, non sono ancora state restituite** e le **informazioni dell’utente che**

possiede attualmente la copia del libro. I recapiti consentono al bibliotecario di mettersi in contatto per sollecitare la restituzione.

Un libro che non è stato prestato nei passati 10 anni viene dismesso dalla biblioteca. Tale operazione viene effettuata dagli amministratori del circuito. Il record ad esso associato non viene eliminato, ma questo non potrà più essere prestato agli utenti. I prestiti in corso restano validi fino alla riconsegna.

Glossario dei Termini

Termine	Descrizione	Sinonimi	Collegamenti
Libro	Tipo di libro presente in biblioteca. La biblioteca ha una o più copie di uno stesso libro.		Copia
Copia	Ogni copia di un libro presente in biblioteca. Può essere prestato a un utente.		Libro, Utente
Utente	Una persona che prende in prestito libri dalla biblioteca		Copia
Bibliotecario	Una persona che lavora in una biblioteca e concede i prestiti agli utenti		

Raggruppamento dei requisiti in insiemi omogenei

Frasi relative a Libro
<p>Un circuito di biblioteche mette a disposizione un servizio di prenotazione libri. [...]</p> <p>Il sistema gestisce libri, le cui copie sono disponibili in un sottoinsieme delle biblioteche, in numero differente. [...]</p> <p>All'atto di effettuare un prestito, i bibliotecari possono recuperare la disponibilità delle copie del libro presso la biblioteca in cui essi lavorano. [...]</p> <p>Un libro che non è stato prestato nei passati 10 anni viene dismesso dalla biblioteca. Tale operazione viene effettuata dagli amministratori del circuito. [...]</p>
Frasi relative a Copia
<p>[...] Il sistema gestisce libri, le cui copie sono disponibili in un sottoinsieme delle biblioteche, in numero differente. Ciascuna copia è associata allo stato "in prestito" / "disponibile". [...]</p> <p>All'atto di effettuare un prestito, i bibliotecari possono recuperare la disponibilità delle copie del libro presso la biblioteca in cui essi lavorano. Se una copia è disponibile, il sistema restituisce lo scaffale ed il ripiano in cui può essere prelevata una copia. Se non ne è disponibile alcuna copia, il bibliotecario può verificare in quali altre biblioteche del circuito è presente tale libro ed effettuare una richiesta di trasferimento. In questo caso, la copia viene segnalata come "prestato ad altra biblioteca" e viene tenuto traccia di qual è la biblioteca di partenza e di destinazione. All'atto della consegna della copia, l'utente del servizio può chiedere di trattenere in consultazione in libro per 1, 2, o 3 mesi.</p> <p>I bibliotecari hanno la possibilità di generare un report indicante quali copie, prestate dalla biblioteca in cui stanno svolgendo il turno, non sono ancora state restituite e le informazioni dell'utente che possiede attualmente la copia del libro. [...]</p>

Frase relative a Utente

[...] Gli utenti del circuito di biblioteche possono registrarsi fornendo tutte le loro informazioni anagrafiche ed un numero arbitrario di contatti (telefono, cellulare, e-mail), specificando qual è il mezzo di comunicazione preferito con cui vogliono essere contattati. [...] All'atto della consegna della copia, l'utente del servizio può chiedere di trattenere in consultazione il libro per 1, 2, o 3 mesi.

I bibliotecari hanno la possibilità di generare un report indicante quali copie, prestate dalla biblioteca in cui stanno svolgendo il turno, non sono ancora state restituite e le informazioni dell'utente che possiede attualmente la copia del libro. [...]

Il record ad esso associato non viene eliminato, ma questo non potrà più essere prestato agli utenti. [...]

Frase relative a Bibliotecario

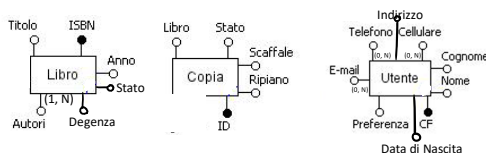
[...] All'atto di effettuare un prestito, i bibliotecari possono recuperare la disponibilità delle copie del libro presso la biblioteca in cui essi lavorano. [...]

I bibliotecari hanno la possibilità di generare un report indicante quali copie, prestate dalla biblioteca in cui stanno svolgendo il turno, non sono ancora state restituite e le informazioni dell'utente che possiede attualmente la copia del libro. I recapiti consentono al bibliotecario di mettersi in contatto per sollecitare la restituzione. [...]

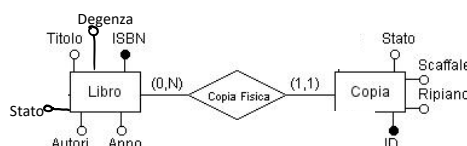
3. Progettazione concettuale

Costruzione dello schema E-R

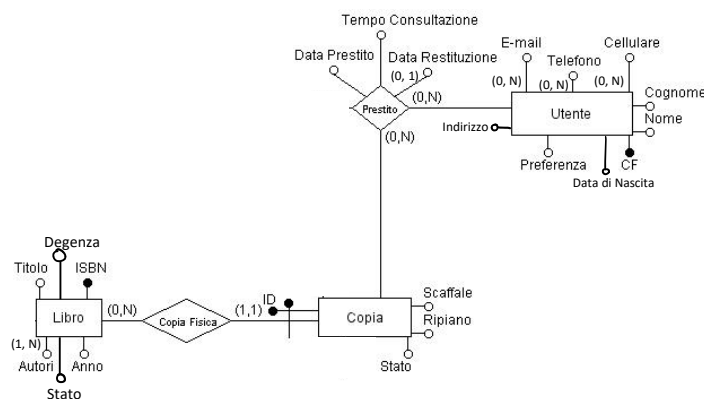
1. Ho iniziato definendo le entità individuate.



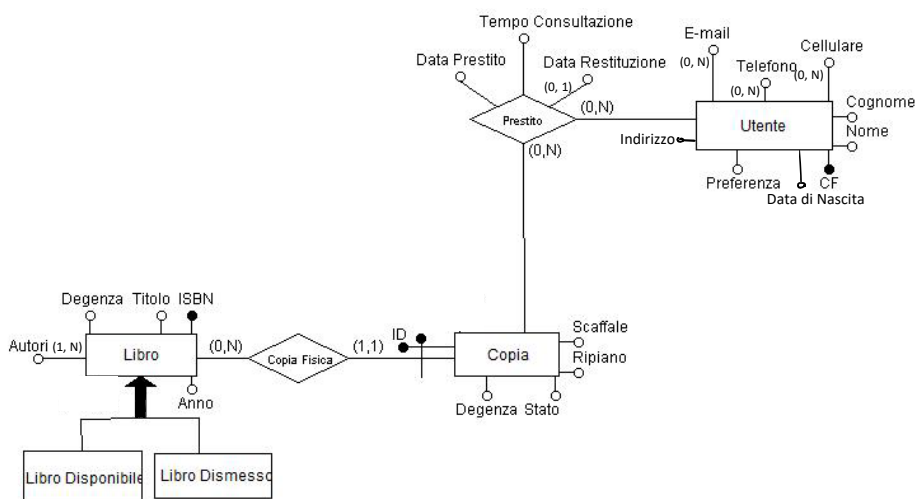
2. Quindi ho iniziato ad associare l'entità "libro" con "copia" con l'associazione "Copia Fisica".



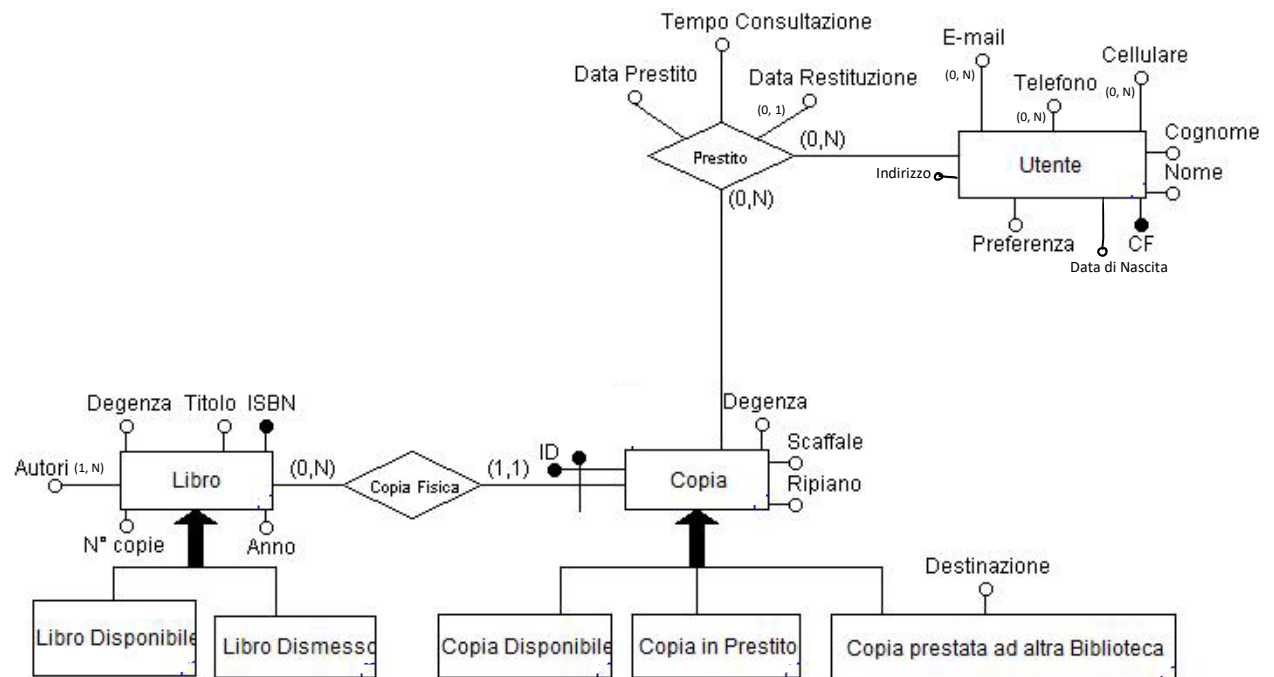
3. Ho associato i due schemi con l'associazione "prestito" tra "copia" e "utente".



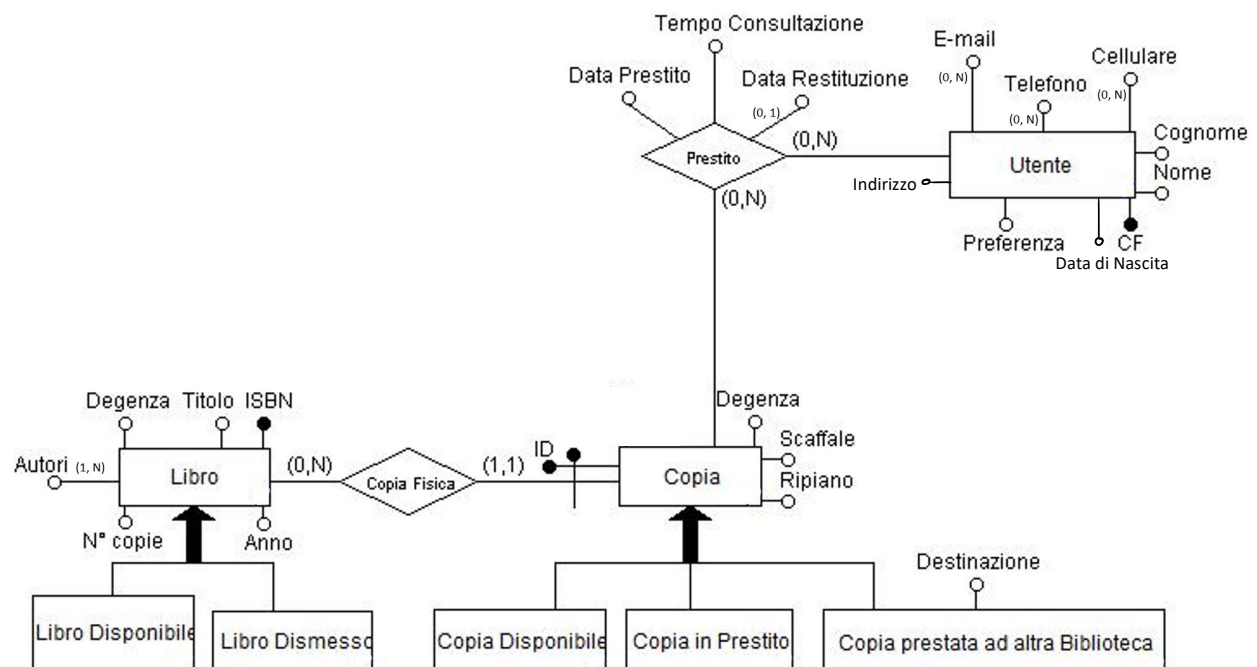
4. Ho reso l'attributo "Stato" di "Libro" una generalizzazione tra "Libro dismesso" e "Libro disponibile".



5. Ho reso l'attributo "Stato" di "Copia" una generalizzazione tra "Copia Disponibile", "Copia in Prestito" e "Copia prestata ad altra Biblioteca" dove l'ultima entità possiede anche l'attributo "Destinazione".



Integrazione finale



Regole aziendali

1. L'utente deve avere almeno un contatto tra telefono, e-mail e cellulare

Dizionario dei dati

Entità	Descrizione	Attributi	Identificatori
Libro	Tipo di libro presente in biblioteca. La biblioteca ha una o più copie di uno stesso libro.	Titolo, Anno, Autori (1, N), ISBN, Degenza, Stato	ISBN
Libro dismesso	Libro dismesso e non prestabile all'utente		ISBN
Libro disponibile	Libro prestabile all'utente del sistema		ISBN
Copia	Ogni copia di un libro presente in biblioteca. Può essere prestato a un utente.	Stato, Ripiano, Scaffale, ID, Copia Fisica	ID, Copia Fisica
Copia Disponibile	Ogni copia di un libro disponibile al prestito in biblioteca.		ID, Copia Fisica
Copia in Prestito	Ogni copia di un libro attualmente in prestito.		ID, Copia Fisica
Copia prestata ad altra Biblioteca	Ogni copia di un libro in prestito ad altra biblioteca.	Destinazione	ID, Copia Fisica
Utente	Una persona che prende in prestito libri dalla biblioteca	Preferenza, E-mail (0, N), Telefono (0, N), Cellulare (0, N), Nome, Cognome, CF, Data di Nascita, Indirizzo	CF

4. Progettazione logica

Volume dei dati

Concetto nello schema	Tipo ¹	Volume atteso
Libro	E	20.000
Libro dismesso	E	1.000
Libro disponibile	E	19.000
Copia	E	80.000
Copia disponibile	E	40.000
Copia in prestito	E	30.000
Copia prestata ad altra biblioteca	E	10.000
Utente	E	1.200
Copia Fisica	R	80.000
Prestito	R	40.000

Riferimenti

AIB. (s.d.). *AIB. Standard per le biblioteche pubbliche (1965)*. Tratto da AIB-WEB – Il web dell'Associazione Italiana Biblioteche:
<https://www.aib.it/aib/stor/testi/stan1965.htm>

Istat. (2021, Aprile 23). *Le biblioteche in italia - Anno 2019 - Istat*. Tratto il giorno 16, 2023 da
https://www.istat.it/it/files//2021/04/REPORT_BIBLIOTECHE-IN-ITALIA.pdf

Tavola delle operazioni

Cod.	Descrizione	Frequenza attesa
OP1	Recuperare la disponibilità delle copie del libro presso la biblioteca	30/giorno
OP2	Dismettere un libro che non è stato prestato nei passati 10 anni.	1/settimana
OP3	Cambiare lo stato della copia (“in prestito” / “disponibile” / “prestato ad altra biblioteca”).	25/giorno
OP4	Generare un report indicante quali copie, prestate non sono ancora state restituite e le informazioni dell’utente che possiede attualmente la copia del libro.	1/giorno
OP5	Prestare un libro ad un utente.	20/giorno
OP6	Registrazione nuovo utente.	5/giorno

¹ Indicare con E le entità, con R le relazioni

Costo delle operazioni

OP1

Recuperare la disponibilità delle copie del libro

- Copia disponibile: 40.000

Costo totale: 40.000

Accessi/giorno: 1.200.000

Concetto	Costrutto	Accessi	Tipo
Copia disponibile	Entità	1	L

OP2

Dismettere un libro non prestato nei passati 10 anni

- Libro disponibile: 60.000
- Copia Fisica: 320.000
- Copia: 320.000

Costo totale: 700.000

Accessi/giorno: 100.000

Concetto	Costrutto	Accessi	Tipo
Libro Disponibile	Entità	1	L
Libro Disponibile	Entità	2	S
Copia Fisica	Associazione	4	S
Copia	Entità	4	L

OP3

Cambiare “*stato*” Copia

- Copia: 160.000

Costo totale: 160.000

Accessi/giorno: 4.000.000

Concetto	Costrutto	Accessi	Tipo
Copia	Entità	2	S

OP4

Generare un report su copie in prestito e utente

- Prestito: N (in media 40.000)
- Utente: 1.200

Costo totale: $1.200+N$ (in media 41.200)

Accessi/giorno: $1.200+N$ (in media 41.200)

Concetto	Costrutto	Accessi	Tipo
Prestito	Associazione	1	L
Utente	Entità	1	L

N è il numero di prestiti attivi.

OP5

Prestare un libro

- Copia: 80.000
- Copia disponibile: 80.000
- Copia Fisica: 80.000
- Libro: 40.000
- Prestito: 120.000
- Utente: 1.200

Costo totale: 401.200

Accessi/giorno: 8.024.000

Concetto	Costrutto	Accessi	Tipo
Copia	Entità	1	L
Copia disponibile	Entità	2	S
Copia Fisica	Associazione	1	L
Libro	Entità	1	S
Prestito	Associazione	1	L
Utente	Entità	1	L
Prestito	Associazione	2	S

OP6

Registrazione nuovo utente

- Utente: 2.400

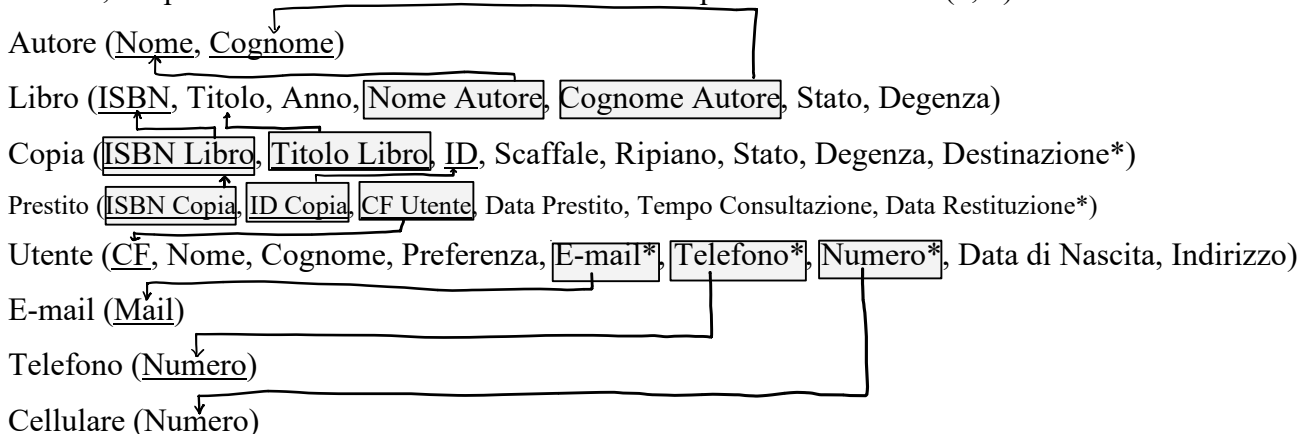
Costo totale: 2.400

Accessi/giorno: 12.000

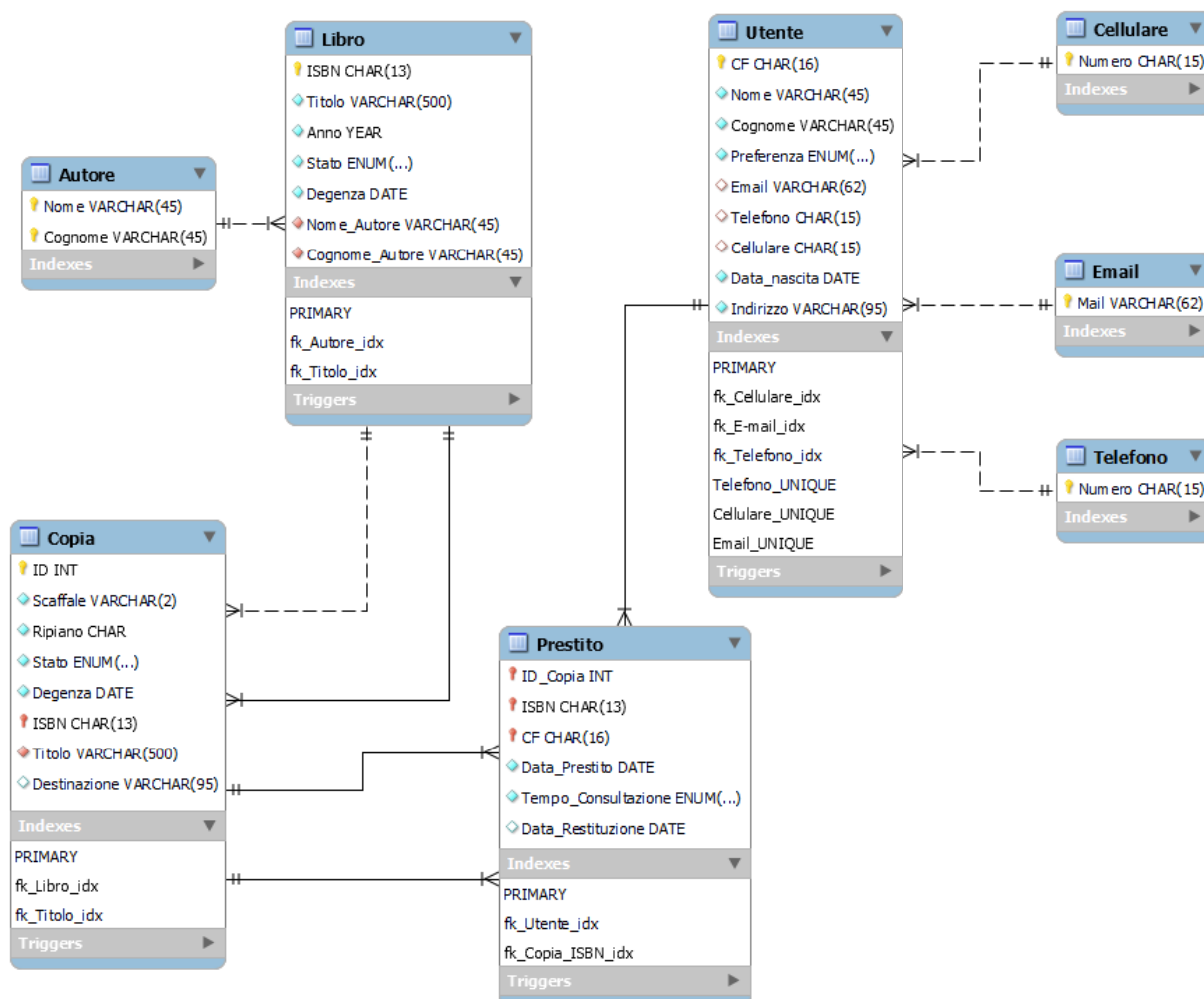
Concetto	Costrutto	Accessi	Tipo
Utente	Entità	2	S

Ristrutturazione dello schema E-R

Nella ristrutturazione ho reso gli attributi multi-valore come relazioni con entità e a parte la relazione “*Prestito*”, trasformata in entità debole, ho unito le relazioni nell’entità con partecipazione a cardinalità minore. Inoltre, le generalizzazioni di Libro e Copia l’ho trasformate in un attributo “*Stato*” di tipo ENUM, trasportando l’attributo “*Destinazione*” su Copia con cardinalità (0, 1).



Traduzione di entità e associazioni



Normalizzazione del modello relazionale

Il modello non è in forma normale. Anomalie di aggiornamento, inserimento ed eliminazione sulla relazione Prestito e l'entità Copia.

Tuttavia, dato il significativo risparmio prestazionale e la tipologia di operazioni previste, si decide di mantenere il modello dei dati in questa forma.

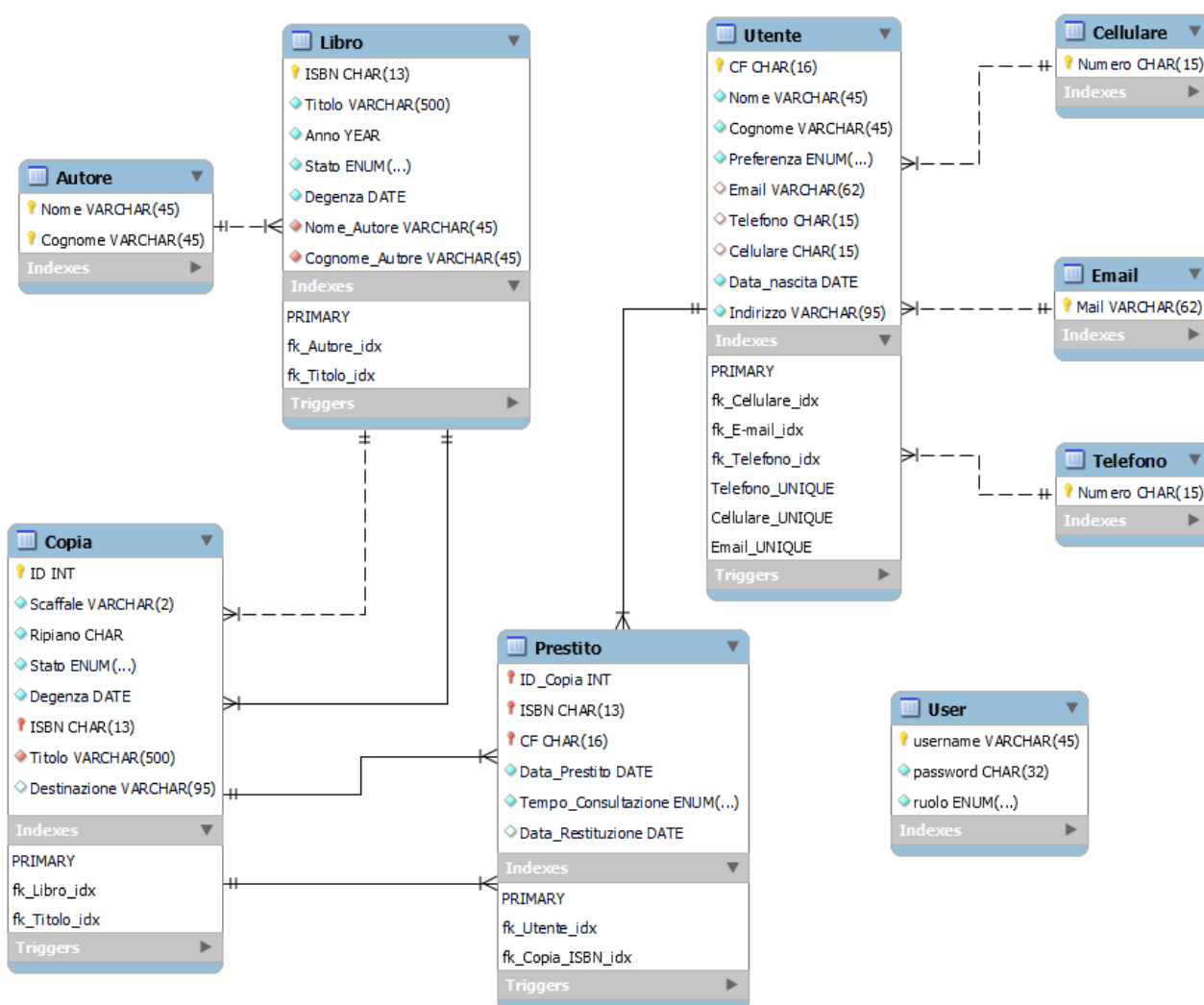
5. Progettazione fisica

Utenti e privilegi

Si prevedono tre ruoli, per implementare il Principle of Least Privilege:

- Bibliotecario:
 - Grant in esecuzione sulle operazioni OP1, OP3, OP4, OP5, OP6
- Amministratore:
 - Grant in esecuzione sulle operazioni OP2

Per identificare gli utenti si introduce una tabella User per mantenerne le credenziali.



Strutture di memorizzazione

Tabella User		
Colonna	Tipo di dato	Attributi ²
username	VARCHAR (45)	PK, NN
password	CHAR (32)	NN
ruolo	ENUM ('bibliotecario', 'amministratore')	NN

Tabella Autore		
Colonna	Tipo di dato	Attributi ²
Nome	VARCHAR (45)	PK, NN
Cognome	VARCHAR (45)	PK, NN

Tabella Libro		
Colonna	Tipo di dato	Attributi ²
ISBN	CHAR (13)	PK, NN
Titolo	VARCHAR (500)	NN
Anno	YEAR	NN
Nome_Autore	VARCHAR (45)	NN
Cognome_Autore	VARCHAR (45)	NN
Stato	ENUM ('disponibile', 'dismesso')	NN
Degenza	DATE	NN

Tabella Telefono		
Colonna	Tipo di dato	Attributi ²
Numero	CHAR (15)	PK, NN

Tabella Email		
Colonna	Tipo di dato	Attributi ²
Mail	VARCHAR (62)	PK, NN

² PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

Tabella Copia		
Colonna	Tipo di dato	Attributi ²
ID	INT	PK, NN, AI
Scaffale	VARCHAR (2)	NN
Ripiano	CHAR	NN
Stato	ENUM ('in prestito', 'disponibile', 'prestato ad altra biblioteca')	NN
Degenza	DATE	NN
ISBN	CHAR (13)	PK, NN
Titolo	VARCHAR (500)	NN
Destinazione	VARCHAR (95)	

Tabella Prestito		
Colonna	Tipo di dato	Attributi ²
ISBN	CHAR (13)	PK, NN
ID_Copia	INT	PK, NN
CF	CHAR (16)	PK, NN
Data_Prestito	DATE	NN
Tempo_Consultazione	ENUM ('1 mese', '2 mesi', '3 mesi')	NN
Data_Restituzione	DATE	

Tabella Utente		
Colonna	Tipo di dato	Attributi ²
CF	CHAR (16)	PK, NN
Nome	VARCHAR (45)	NN
Cognome	VARCHAR (45)	NN
Preferenza	ENUM ('telefono', 'cellulare', 'e-mail')	NN
Email	VARCHAR (62)	
Telefono	CHAR (15)	
Cellulare	CHAR (15)	
Data_nascita	DATE	NN
Indirizzo	VARCHAR (95)	NN

Tabella Cellulare		
Colonna	Tipo di dato	Attributi ²
Numero	CHAR (15)	PK, NN

Indici

Considerando una possibile ricerca per Titolo su Copia e che la tabella Prestito è debole verso Copia che a sua volta è debole verso Libro, si è considerato più ottimale aggiungere un indice verso Titolo della Tabella Libro per riportarlo su Copia piuttosto che inserirlo nella Primary Key di Libro.

Tabella Libro	
Indice fk_Titolo_idx	Tipo ³ :
Titolo	IDX

Inoltre si è ritenuto necessario mettere un indice unique sui possibili contatti dell'utente.

Tabella Utente	
Indice Telefono_UNIQUE	Tipo ³ :
Telefono	UQ
Indice Cellulare_UNIQUE	Tipo ³ :
Cellulare	UQ
Indice Email_UNIQUE	Tipo ³ :
Email	UQ

Trigger

Tabella Utente: BEFORE INSERT

Il seguente trigger implementa un controllo per evitare che l'indirizzo sia nel formato "strada, numero, CAP, città" (ad esempio "Via Roma, 12, 00100, Roma").

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Check_address` BEFORE INSERT ON
`Utente` FOR EACH ROW
BEGIN
    IF NOT REGEXP_LIKE(NEW.`Indirizzo`, '^[A-Za-z ]+, [0-9]+, [0-9]{5}, [A-Za-z ]+$')
    THEN
        SIGNAL SQLSTATE '45001' SET message_text = 'Invalid address format';
    END IF;
END
```

Tabella Utente: BEFORE INSERT

Il seguente trigger implementa un controllo per evitare che il codice fiscale sia nel formato errato.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Check_CF` BEFORE INSERT ON `Utente` FOR
EACH ROW
BEGIN
    IF NEW.`CF` NOT REGEXP '^[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]$' THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Codice fiscale non valido';
    END IF;
END
```

³ IDX = index, UQ = unique, FT = full text, PR = primary.

Tabella Utente: BEFORE INSERT

Il seguente trigger implementa un controllo per evitare che si inserisca un utente senza recapito.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Check_Contact` BEFORE INSERT ON
`Utente` FOR EACH ROW
BEGIN
    IF (NEW.`Email` IS NULL AND NEW.`Telefono` IS NULL AND NEW.`Cellulare` IS NULL)
    THEN
        SIGNAL SQLSTATE '45001' SET message_text = 'Inserire almeno un contatto!';
    END IF;
END
```

Tabella Utente: BEFORE INSERT

Il seguente trigger implementa un controllo per evitare che si inserisca un utente con il recapito preferito a null.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Check_Favorite_Contact` BEFORE INSERT
ON `Utente` FOR EACH ROW
BEGIN
    IF (NEW.`Email` IS NULL AND NEW.`Preferenza` = 'e-mail') THEN
        SIGNAL SQLSTATE '45001' SET message_text = 'Inserire il contatto
preferito!';
    ELSEIF (NEW.`Telefono` IS NULL AND NEW.`Preferenza` = 'telefono') THEN
        SIGNAL SQLSTATE '45001' SET message_text = 'Inserire il contatto
preferito!';
    ELSEIF (NEW.`Cellulare` IS NULL AND NEW.`Preferenza` = 'cellulare') THEN
        SIGNAL SQLSTATE '45001' SET message_text = 'Inserire il contatto
preferito!';
    END IF;
END
```

Tabella Utente: BEFORE INSERT

Il seguente trigger implementa l'aggiunta della nuova mail, numero di telefono o cellulare nel caso non esista già nella Base di Dati.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Add_Contact` BEFORE INSERT ON `Utente`
FOR EACH ROW
BEGIN
    IF (NEW.`Email` IS NOT NULL) THEN
        IF NOT EXISTS (SELECT `Mail` FROM `Email` WHERE `Mail` = NEW.`Email`) THEN
            INSERT INTO `Email` VALUE (NEW.`Email`);
        END IF;
    END IF;

    IF (NEW.`Telefono` IS NOT NULL) THEN
        IF NOT EXISTS (SELECT `Numero` FROM `Telefono` WHERE `Numero` =
NEW.`Telefono`) THEN
            INSERT INTO `Telefono` VALUE (NEW.`Telefono`);
        END IF;
    END IF;

    IF (NEW.`Cellulare` IS NOT NULL) THEN
        IF NOT EXISTS (SELECT `Numero` FROM `Cellulare` WHERE `Numero` =
NEW.`Cellulare`) THEN
            INSERT INTO `Cellulare` VALUE (NEW.`Cellulare`);
        END IF;
    END IF;
END
```

Tabella Utente: BEFORE INSERT

Il seguente trigger implementa l'aggiunta della nuova mail, numero di telefono o cellulare nel caso non esista già nella Base di Dati.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`utenteIsExist` BEFORE INSERT ON
`Utente` FOR EACH ROW
BEGIN
    DECLARE var_iscritto INT;

    SELECT COUNT(*)
    FROM `Utente`
    WHERE `CF` = NEW.`CF`
    INTO var_iscritto;

    IF var_iscritto > 0 THEN
        SIGNAL SQLSTATE '45001' SET message_text = "The user is already
registered!";
    END IF;
END
```

Tabella Copia: BEFORE UPDATE

Il seguente trigger implementa un controllo per evitare che si inserisca una copia “prestata ad altra biblioteca” senza biblioteca di destinazione e verifica che l'indirizzo sia nel formato "strada, numero, CAP, città" (ad esempio "Via Roma, 12, 00100, Roma").

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Check_Destination` BEFORE UPDATE ON
`Copia` FOR EACH ROW
BEGIN
    IF (NEW.`Stato` = 'prestato ad altra biblioteca' AND NEW.`Destinazione` IS NULL)
    THEN
        SIGNAL SQLSTATE '45001' SET message_text = 'Inserire biblioteca di
destinazione!';
        END IF;

    IF (NEW.`Stato` = 'prestato ad altra biblioteca' AND NEW.`Destinazione` IS NOT
NULL) THEN
        IF NOT REGEXP_LIKE(NEW.`Destinazione`, '^[A-Za-z ]+, [0-9]+, [0-9]{5}, [A-
Za-z ]+$') THEN
            SIGNAL SQLSTATE '45001' SET message_text = 'Invalid address format';
            END IF;
        END IF;
    END
```

Tabella Copia: BEFORE INSERT

Il seguente trigger implementa nel caso in cui non si specifichi la data di degenza, l'aggiunta della data attuale e l'aggiornamento su Libro.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Set_Copia_Current_Date` BEFORE INSERT
ON `Copia` FOR EACH ROW
BEGIN
    IF (NEW.`Degenza` IS NULL) THEN
        SET NEW.`Degenza` = CURRENT_DATE();
        END IF;

    IF (NEW.`Degenza` IS NOT NULL) THEN
        IF NOT EXISTS (SELECT `ISBN`, `Titolo` FROM `Copia` WHERE `ISBN` =
NEW.`ISBN` AND `Titolo` = NEW.`Titolo`) THEN
            UPDATE `Libro`
            SET `Degenza` = NEW.`Degenza`
            WHERE `ISBN` = NEW.`ISBN` AND `Titolo` = NEW.`Titolo`;
            END IF;
        END IF;
    END
```

Tabella Copia: AFTER UPDATE

Il seguente trigger aggiorna la degenza del Libro dopo aver fatto l'Update di Copia.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Update_Degenza_Libro` AFTER UPDATE ON
`Copia` FOR EACH ROW
BEGIN
    UPDATE `Libro`
    SET `Degenza` = NEW.`Degenza`
    WHERE `ISBN` = NEW.`ISBN`
    AND `Titolo` = NEW.`Titolo`;
    END
```

Tabella Prestito: BEFORE INSERT

Il seguente trigger aggiunge un controllo sull'esistenza della copia da prestare.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`copyIsExist` BEFORE INSERT ON
`Prestito` FOR EACH ROW
BEGIN
    DECLARE var_copia INT;
    SELECT COUNT(*) INTO var_copia
    FROM `Copia`
    WHERE `ISBN` = NEW.`ISBN` AND `ID` = NEW.`ID_Copia`;

    IF var_copia = 0 THEN
        SIGNAL SQLSTATE '45001' SET message_text = "The copy does not
exist!";
    END IF;
END
```

Tabella Prestito: BEFORE INSERT

Il seguente trigger aggiunge un controllo sull'esistenza dell'utente a cui fare il prestito.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`userIsExist` BEFORE INSERT ON
`Prestito` FOR EACH ROW
BEGIN
    DECLARE var_iscritto INT;
    SELECT COUNT(*) INTO var_iscritto
    FROM `Utente`
    WHERE `CF` = NEW.`CF`;
    IF var_iscritto = 0 THEN
        SIGNAL SQLSTATE '45001' SET message_text = "The user is not
registered!";
    END IF;
END
```

Tabella Prestito: AFTER INSERT

Il seguente trigger aggiorna la data della degenza della Copia prestata.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Update_Degenza_Prestito` AFTER INSERT
ON `Prestito` FOR EACH ROW
BEGIN
    UPDATE `Copia`
    SET `Degenza` = NEW.`Data_Prestito`
    WHERE `ISBN` = NEW.`ISBN`
    AND `ID` = NEW.`ID_Copia`;
END
```

Tabella Prestito: AFTER INSERT

Il seguente trigger aggiorna lo stato della Copia prestata in "in prestito".

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`updateCopy` AFTER INSERT ON `Prestito`
FOR EACH ROW
BEGIN
    UPDATE `Copia`
    SET `Stato` = 'in prestito'
    WHERE `ISBN` = NEW.`ISBN`
    AND `ID` = NEW.`ID_Copia`;
END
```

Tabella Libro: BEFORE INSERT

Il seguente trigger implementa l'aggiunta di un nuovo Autore nel caso non esista già nella Base di Dati.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Add_Author` BEFORE INSERT ON `Libro`  
FOR EACH ROW  
BEGIN  
    IF (NEW.`Nome_Autore` IS NOT NULL AND NEW.`Cognome_Autore` IS NOT NULL) THEN  
        IF NOT EXISTS (SELECT `Nome`, `Cognome` FROM `Autore` WHERE `Nome` =  
NEW.`Nome_Autore` AND `Cognome` = NEW.`Cognome_Autore`) THEN  
            INSERT INTO `Autore` VALUE (NEW.`Nome_Autore`,  
NEW.`Cognome_Autore`);  
        END IF;  
    END IF;  
END
```

Tabella Libro: BEFORE INSERT

Il seguente trigger implementa un controllo per evitare che l'ISBN sia nel formato errato.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Check_ISBN` BEFORE INSERT ON `Libro`  
FOR EACH ROW  
BEGIN  
    IF NEW.`ISBN` NOT REGEXP '[0-9]{13}' THEN  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'ISBN non valido';  
    END IF;  
END
```

Tabella Libro: BEFORE INSERT

Il seguente trigger implementa nel caso in cui non si specifichi la data di degenza, l'aggiunta della data attuale.

```
CREATE DEFINER = CURRENT_USER TRIGGER `Library`.`Set_Libro_Current_Date` BEFORE INSERT  
ON `Libro` FOR EACH ROW  
BEGIN  
    IF (NEW.`Degenza` IS NULL) THEN  
        SET NEW.`Degenza` = CURRENT_DATE();  
    END IF;  
END
```

Eventi

Non sono stati utilizzati eventi.

Viste

Non sono state utilizzate viste

Stored Procedures e transazioni

Operazione Login:

```
CREATE PROCEDURE `login` (IN var_username VARCHAR(45), IN var_pass CHAR(32), OUT var_role
INT)
BEGIN
    DECLARE userExists INT;
    DECLARE var_user_role ENUM('bibliotecario', 'amministratore');

    SELECT COUNT(*) FROM `User`
    WHERE `username` = var_username
    AND `password` = md5(var_pass)
    INTO userExists;

    SELECT `ruolo` from `User`
    WHERE `username` = var_username
    AND `password` = md5(var_pass)
    INTO var_user_role;

    IF (userExists = 1) THEN
        SELECT 'Access granted.' AS 'Message';
        IF var_user_role = 'bibliotecario' THEN
            SET var_role = 1;
        ELSEIF var_user_role = 'amministratore' THEN
            SET var_role = 2;
        END IF;
    ELSE
        SELECT 'Access denied.' AS 'Message';
        SET var_role = 3;
    END IF;
END
```

Operazione Crea User:

```
CREATE PROCEDURE `crea_user` (IN username VARCHAR(45), IN pass VARCHAR(45), ruolo
VARCHAR(45))
BEGIN
    INSERT INTO `User` VALUES (username, MD5(pass), ruolo);
END
```


Operazione OP1:

```
CREATE PROCEDURE `get_book_availability` (IN var_title VARCHAR(500), IN var_ISBN  
CHAR(13))
```

```
BEGIN
```

```
    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

```
    START TRANSACTION;
```

```
        IF var_ISBN <> '' THEN
```

```
            SELECT * FROM `Copia`
```

```
            WHERE `ISBN` = var_ISBN
```

```
            AND `Stato` = 'disponibile';
```

```
        ELSEIF var_title <> '' THEN
```

```
            SELECT * FROM `Copia`
```

```
            WHERE `Titolo` = var_title
```

```
            AND `Stato` = 'disponibile';
```

```
        ELSE
```

```
            SELECT * FROM `Copia`
```

```
            WHERE `Stato` = 'disponibile';
```

```
        END IF;
```

```
    COMMIT;
```

```
END
```

Operazione OP3:

```
CREATE PROCEDURE `change_copia_state` (IN var_ID INT, IN var_ISBN CHAR(13), IN var_state  
VARCHAR(45))
```

```
BEGIN
```

```
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
```

```
    BEGIN
```

```
        ROLLBACK;
```

```
        RESIGNAL;
```

```
    END;
```

```
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

```
    START TRANSACTION;
```

```
        UPDATE `Copia` SET `Stato` = var_state WHERE `ISBN` = var_ISBN AND `ID` =  
var_ID;
```

```
    COMMIT;
```

```
END
```

Operazione OP2:

```
CREATE PROCEDURE `remove_book` ()
BEGIN
    DECLARE var_ISBN CHAR(13);
    DECLARE done INT DEFAULT FALSE;
    DECLARE cur CURSOR FOR SELECT `ISBN` FROM `Libro` WHERE `Degenza` <
DATE_SUB(NOW(), INTERVAL 10 YEAR);
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
    START TRANSACTION;
    OPEN cur;
    read_loop: LOOP
        FETCH cur INTO var_ISBN;
        IF done THEN
            LEAVE read_loop;
        END IF;

        DELETE FROM `Copia`
        WHERE `ISBN` = var_ISBN;

        UPDATE `Libro`
        SET `Stato` = 'dismesso'
        WHERE `ISBN` = var_ISBN;
    END LOOP;
    CLOSE cur;
    COMMIT;
END
```

Operazione OP4:

```
CREATE PROCEDURE `report_prestito` ()
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    DROP TEMPORARY TABLE IF EXISTS `active_booking`;
    CREATE TEMPORARY TABLE `active_booking` (
        `Data_Prestito` DATE,
        `Tempo_Consultazione` ENUM('1 mese', '2 mesi', '3 mesi'),
        `ISBN` CHAR(13), `ID` INT,
        `Nome` VARCHAR(45),
        `Cognome` VARCHAR(45),
        `CF` CHAR(16), `Email` VARCHAR(62),
        `Telefono` CHAR(15), `Cellulare` CHAR(15),
        `Preferenza` ENUM('telefono', 'cellulare', 'e-mail')
    );

    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
    START TRANSACTION;

    INSERT INTO `active_booking`
    SELECT `Data_Prestito`, `Tempo_Consultazione`, `Prestito`.`ISBN`, `ID`,
        `Nome`, `Cognome`, `Prestito`.`CF`, `Email`, `Telefono`,
        `Cellulare`, `Preferenza`
    FROM `Prestito`
    JOIN `Utente` ON `Prestito`.`CF` = `Utente`.`CF`
    JOIN `Copia` ON `Prestito`.`ISBN` = `Copia`.`ISBN`
    AND `Prestito`.`ID_Copia` = `Copia`.`ID`
    WHERE `Prestito`.`Data_Restituzione` IS NULL
    ORDER BY `Data_Prestito` ASC;

    SELECT * FROM `active_booking`;

    COMMIT;
    DROP TEMPORARY TABLE IF EXISTS `active_booking`;
END
```

Operazione OP5:

```
CREATE PROCEDURE `booking_book` (IN var_ID INT, IN var_ISBN CHAR(13), IN var_CF  
CHAR(16), IN var_temp ENUM('1 mese', '2 mesi', '3 mesi'))  
BEGIN  
    DECLARE EXIT HANDLER FOR SQLEXCEPTION  
    BEGIN  
        ROLLBACK;  
        RESIGNAL;  
    END;  
  
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
    START TRANSACTION;  
        INSERT INTO `Prestito` (ID_Copia, ISBN, CF, Data_Prestito,  
Tempo_Consultazione) VALUES (var_ID, var_ISBN, var_CF, CURRENT_DATE(), var_temp);  
    COMMIT;  
END
```

Operazione OP6:

```
CREATE PROCEDURE `iscrivi_utente` (IN var_cf CHAR(16), IN var_name VARCHAR(45), IN  
var_surname VARCHAR(45), IN var_pref VARCHAR(45), IN var_mail VARCHAR(62), IN var_phone  
VARCHAR(45), IN var_cell VARCHAR(45), IN var_bday DATE, IN var_address VARCHAR(95))  
BEGIN  
    DECLARE EXIT HANDLER FOR SQLEXCEPTION  
    BEGIN  
        ROLLBACK;  
        RESIGNAL;  
    END;  
  
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
    START TRANSACTION;  
        INSERT INTO `Utente` VALUES (var_cf, var_name, var_surname, var_pref,  
var_mail, var_phone, var_cell, var_bday, var_address);  
    COMMIT;  
END
```