

Progetto db_ebiblio

Tiziano Bruno(883120), Marco Carnevali(873405)

2021



Figure 1: Biblioteca Universitaria di Bologna

Indice

1 Raccolta e Analisi dei requisiti	3
1.1 Testo completo della specifica sui dati	3
1.1.1 Premessa	3
1.1.2 Specifica	3
1.2 Lista delle operazioni	5
1.3 Tavola media dei volumi	7
1.4 Glossario dei dati	8
2 Progettazione Concettuale	9
2.1 Diagramma E-R	9
2.2 Dizionario delle entità/relazioni	10
2.3 Tavola delle bussiness rules	12
3 Progettazione Logica	13
3.1 Ristrutturazione dello schema concettuale	13
3.2 Analisi delle ridondanze	13
3.3 Lista delle tabelle	15
3.4 Lista dei vincoli inter-relazionali	15
4 Descrizione (ad alto livello) delle funzionalità dell'applicazione web	17
4.1 Istruzioni di avvio	17
4.1.1 Prerequisiti	17
4.1.2 Setup	18
4.2 Avvio	18
5 Clustering utenti utilizzatori	19
5.1 K-Means con due cluster	20
5.2 K-Means con tre cluster	21
5.3 K-Means con quattro cluster	22
5.4 Conclusioni sulla clusterizzazione	24
A Codice SQL completo dello schema della base di dati	25
A.1 Creazione delle tabelle	25
A.2 Operazioni che riguardano tutti gli utenti	29
A.3 Operazioni che riguardano solo gli utenti UTILIZZATORI	32
A.4 Operazioni che riguardano solo gli utenti VOLONTARI	35
A.5 Operazioni che riguardano solo gli utenti AMMINISTRATORI	38
A.6 Classifiche	42
Indice	

List of Tables

1	Tabella media dei volumi db_ebiblio	7
2	Glossario dei dati db_ebiblio	8
3	Dizionario delle entità db_ebiblio	10
4	Dizionario delle relazioni db_ebiblio	11

1 Raccolta e Analisi dei requisiti

1.1 Testo completo della specifica sui dati

1.1.1 Premessa

Si vuole realizzare la piattaforma EBIBLIO, una piattaforma per la gestione di biblioteche UNIBO durante il periodo della pandemia COVID-19. La piattaforma gestisce un database di libri (cartacei ed e-book) disponibili presso le biblioteche UNIBO, e supporta un meccanismo di prenotazione e consegna dei testi a domicilio, oltre che di prenotazione dei posti contingentati presso le aule studio di ciascuna biblioteca. I libri sono recapitati da un insieme ristretto di utenti volontari. Gli utenti utilizzatori delle biblioteche possono accedere -tramite la piattaforma- agli e-book, prenotare la consegna di un testo (a domicilio) oppure prenotare l'accesso fisico alle postazioni contingentate. La piattaforma dispone di un servizio interno di messaggistica ed alerting, gestita dagli amministratori delle biblioteche.

1.1.2 Specifica

La piattaforma EBIBLIO deve gestire i dati delle biblioteche UNIBO. Ogni biblioteca dispone di un nome (univoco), un indirizzo, un'email, un sito web, delle coordinate (latitudine/longitudine), uno o più recapiti telefonici, un campo di testo relativo alle note storiche. Ogni biblioteca può disporre di una galleria di immagini (una o più). Inoltre, ogni biblioteca dispone di un numero limitato di posti lettura: ogni posto lettura dispone di un numero progressivo (univoco, ma solo all'interno di una biblioteca), dell'indicazione se dotato di presa di corrente o meno (campo booleano) e dell'indicazione se dotato di presa di rete Ethernet o meno (campo booleano). La biblioteca mette a disposizione del pubblico l'accesso ad i propri libri: ogni libro dispone di un codice (univoco a livello UNIBO), un titolo, una lista degli autori, un anno di pubblicazione, un nome dell'edizione, un genere. I libri possono appartenere a due categorie (e solo a quelli): libri cartacei o ebook. Nel primo caso, si vogliono memorizzare anche stato di conservazione, numero di pagine, numero scaffale, e stato del prestito. Lo stato di conservazione può assumere solo quattro valori: Ottimo, Buono, Non Buono, Scadente. Lo stato del prestito può essere: Disponibile, Prenotato, Consegnato. Nel caso degli e-book, si vogliono memorizzare anche la dimensione, il numero di accessi alla scheda e il PDF del documento. Alla piattaforma EBIBLIO possono accedere tre categorie di utenti: Amministratori, Volontari e Utilizzatori. Ogni utente dispone di email, password, nome, cognome, data di nascita, luogo di nascita, recapito telefonico. Gli utenti possono appartenere a tre categorie: amministratori di biblioteca, volontari e utilizzatori. Gli amministratori (dipendenti UNIBO) dispongono anche del campo qualifica (testo, massimo 10 caratteri); ogni amministratore è responsabile di una sola biblioteca UNIBO. Una biblioteca UNIBO può avere più amministratori. Gli utenti volontari hanno un campo mezzo di trasporto (piedi, bici, auto).

Gli utenti utilizzatori dispongono di un campo aggiuntivo relativo alla data di creazione dell'account, un campo professione, ed un campo relativo allo stato dell'account (quest'ultimo può assumere solo due valori: Attivo o Sospeso). Gli utilizzatori possono prenotare un posto lettura presso una biblioteca UNIBO; ogni prenotazione dispone di un campo data, ora inizio ed ora fine. La prenotazione di un posto lettura è possibile solo a condizione che la biblioteca abbia effettivamente posti lettura disponibili per la data/orario richiesto. Gli utilizzatori possono accedere liberamente agli e-book disponibili: tuttavia, il sistema tiene traccia dello storico degli accessi agli e-book (o meglio alle loro schede) effettuati da ciascun utente. In maniera simile, gli utilizzatori possono prenotare un libro cartaceo, a patto che il testo sia nello stato Disponibile, e che lo stato di conversazione non sia pari a Scadente. La prenotazione dispone di un codice (univoco), una data di avvio e una data di fine (automaticamente settata a +15gg a partire dalla data di consegna, vedi sotto). Gli utenti volontari si fanno carico di consegnare i libri prenotati agli utilizzatori: a tal proposito si vogliono gestire gli eventi di consegna: ogni evento di consegna è inserito da un utente volontario, fa riferimento ad una prenotazione di testo cartaceo, può essere di tipo "Restituzione" o "Affidamento" e dispone di

campo data e note (massimo 200 caratteri). Infine, è prevista la possibilità di gestire messaggi nella piattaforma. Ogni messaggio è inserito da un amministratore ed è destinato ad un utente utilizzatore, e dispone di un titolo (es. “Libro non disponibile”), un campo testo ed una data. Infine, gli amministratori possono inviare segnalazioni per comportamenti non corretti da parte di utilizzatori; ogni segnalazione dispone di data ed eventuale nota di testo, è inserita da un amministratore e diretta verso un utilizzatore. Nel caso in cui un utilizzatore riceva cumulativamente più di tre segnalazioni (anche da amministratori di biblioteche diverse), lo stato dell’account viene settato a “Sospeso” (impedendo qualsiasi accesso alla piattaforma da parte dell’utente sanzionato). *Infine, si*

vuole tenere traccia di tutti gli eventi che occorrono nella piattaforma, relativamente all’inserimento di nuovi dati (es. nuovi utenti, nuovi libri cartacei, etc). Tali eventi vanno inseriti, sotto forma di messaggi di testo, all’interno di un log, implementato in un’ apposita collezione MongoDB.

1.2 Lista delle operazioni

Operazioni che riguardano tutti gli utenti:

- Autenticazione alla piattaforma \implies procedure Autenticazione()
- Visualizzazione delle biblioteche presenti \implies procedure VisualBiblioteche()
- Visualizzazione foto biblioteca nella pagina dedicata \implies procedure FotoBib()
- Visualizzazione informazioni della biblioteca \implies procedure VisualBiblioteca()
- Visualizzazione dei posti lettura presenti in tutte le biblioteche \implies procedure VisualPosti()
- Visualizzazione dei libri disponibili in tutte le biblioteche \implies procedure VisualLibri()
- Visualizzazione dei libri cartacei di una biblioteca \implies procedure VisualCartaceiBib()
- Visualizzazione degli ebook di una biblioteca \implies procedure VisualEbookBib()
- Visualizzazione autori di un libro \implies procedure VisualAutori()
- Visualizzazione delle informazioni di un ebook e incremento del counter degli accessi \implies procedure VisualEbook()
- Visualizzazione numeri di telefono di una biblioteca \implies procedure NumeriTelefono()
- Visualizzazioni delle informazioni di un libro cartaceo \implies procedure CartaceoScelto()

Operazioni che riguardano SOLO gli utenti UTILIZZATORI:

- Registrazione utilizzatore alla piattaforma \implies procedure RegistrazioneUtente()
- Visualizzazione posti lettura disponibili di una biblioteca in un lasso di tempo \implies procedure PostiDisponibili()
- Prenotazione di un posto lettura \implies procedure PrenotazionePosto()
- Visualizzazione delle prenotazioni di posti effettuate da un utente \implies procedure VisualPostiUt()
- Richiesta di prestito di un libro cartaceo \implies procedure PrestitoCartaceo()
- Visualizzazione dei prestiti richiesti da un utente \implies procedure PrestitiUtente()
- Visualizzazione degli eventi di consegna di un utente \implies procedure VisualConsegne()
- Inserimento nello storico Ebook dell'accesso ad un ebook effettuato da un utente \implies procedure StoricoEbook()
- Visualizzazione messaggi ricevuti \implies procedure VisualMessaggi()
- Visualizzazioni segnalazioni ricevute \implies procedure VisualSegnalazioni

Operazioni che riguardano SOLO gli utenti VOLONTARI:

- Registrazione volontario alla piattaforma \implies procedure RegistrazioneVolontario()
- Visualizzazione di tutte le prenotazioni inserite sulla piattaforma \implies procedure VisualPrenotazioniCartacei()

- Inserimento di un nuovo evento di consegna \implies procedure InsertConsegna()
- Aggiornamento di un evento di consegna \implies procedure UpdateConsegna()

Operazioni che riguardano SOLO gli utenti AMMINISTRATORI:

- Return del nome della biblioteca gestita dall'amministratore scelto \implies procedure BibliotecaAmministratore()
- Inserimento degli autori di un libro e collegamento autori-libro \implies procedure Insertautori
- Inserimento/Cancellazione/Aggiornamento di un libro presso la biblioteca gestita \implies procedure InsertLibro(), DeleteLibro(), UpdateLibro()
- Visualizzazione di tutte le prenotazioni posti lettura presso la biblioteca gestita \implies procedure VisualPrenotazioniPosti()
- Inserimento di un messaggio rivolto ad un utente utilizzatore \implies procedure InsertMessaggio()
- Inserimento di una segnalazione di comportamento non corretto \implies procedure InsertSegnalazione()
- Rimuovere tutte le segnalazioni di un utente, riportandone lo stato ad Attivo \implies procedure AssoluzioneUtente()

Statistiche:

- Visualizzare la classifica dei volontari che hanno effettuato più consegne \implies procedure ClassificaVol()
- Visualizzare la classifica dei libri cartacei più prenotati \implies procedure ClassificaCartacei()
- Visualizzare la classifica degli e-book più acceduti \implies procedure ClassificaEbook()
- Visualizzare la classifica delle biblioteche con postazioni letture meno utilizzate (in percentuale rispetto al numero di posti letture disponibili) \implies procedure ClassificaBibliotecheMenoUsate()

1.3 Tavola media dei volumi

Concetto	Tipo	Volume
BIBLIOTECA	Entità	5
TELEFONO	Entità	11
FOTO	Entità	20
LIBRO	Entità	25
LIBRO CARTECEO	Entità	13
LIBRO EBOOK	Entità	12
POSTILETTURA	Entità	15
UTENTE	Entità	21
UTENTE AMMINISTRATORE	Entità	5
UTENTE UTILIZZATORE	Entità	13
UTENTE VOLONTARIO	Entità	3
MESSAGGIO	Entità	2
SEGNALAZIONE	Entità	1
PRENOTAZIONE	Entità	14
CONSEGNA	Entità	6
PRESTITO	Entità	7
RECAPITI	Relazione	5
GALLERIA	Relazione	5
RESPONSABILE	Relazione	5
AULA_STUDIO	Relazione	5
LIBRERIE	Relazione	5
INSERIMENTO	Relazione	25
VIAGGIO	Relazione	6
AFFIDAMENTO	Relazione	6
DOMANDA	Relazione	7
STORICO_EBOOK	Relazione	15
OCCUPAZIONE	Relazione	14
RICHIESTA	Relazione	14
INVIO	Relazione	2
RICEZIONE	Relazione	2
INVIO2	Relazione	1
RICEZIONE2	Relazione	1

Table 1: Tabella media dei volumi db_ebiblio

1.4 Glossario dei dati

Termine	Descrizione	Sinonimi	Collegamenti
Biblioteca	Luogo dove sono raccolti i libri per uso di studio	Libreria, Aula studio	Telefono, Foto, Amministratore responsabile, Posto lettura, Libro
Recapito telefonico	Numero/i di telefono della biblioteca	Telefono	Biblioteca
Galleria immagini	Immagini della biblioteca	Foto	Biblioteca
Libro	Libro fisico o digitale che appartiene alla biblioteca	Cartaceo, E-book	Biblioteca, Prestito, Storico E-book
Autore	Autore di uno o più libri		Libro
Posto lettura	Posto numerato, atto allo studio, presente all'interno della biblioteca		Biblioteca, Prenotazione
Utente	Utente iscritto alla piattaforma		Biblioteca, Messaggio, Segnalazione, Prenotazione, Consegna, Prestito
Amministratore	Utente amministratore che modera la piattaforma e può mandare messaggi. Ogni amministratore è responsabile di una biblioteca	Responsabile	Biblioteca, Messaggio, Segnalazione
Volontario	Utente che si occupa della consegna e restituzione dei libri cartacei		Consegna
Messaggio	Solo gli amministratori possono mandare dei messaggi agli utenti utilizzatori		Amministratore, Utilizzatore
Segnalazione	Solo gli amministratori possono segnalare gli utenti utilizzatori		Amministratore, Utilizzatore
Prenotazione	Ogni utente utilizzatore può prenotare un posto lettura in una biblioteca	Richiesta	Posti lettura, Utilizzatore
Consegna	Gli utenti volontari si occupano di consegnare e ritirare i libri dati in prestito	Viaggio, Affidamento	Volontario, Prestito
Prestito	Ogni utente utilizzatore può prenotare un posto all'interno della biblioteca		Consegna, Utilizzatore, Libro Cartaceo
Responsabile	I responsabili della biblioteca sono amministratori	Amministratore	Biblioteca, Amministratore
Storico e-book	Il sistema tiene traccia degli accessi alle schede degli e-book effettuati da ciascun utente		E-book, Utilizzatore

Table 2: Glossario dei dati db_ebiblio

2 Progettazione Concettuale

2.1 Diagramma E-R

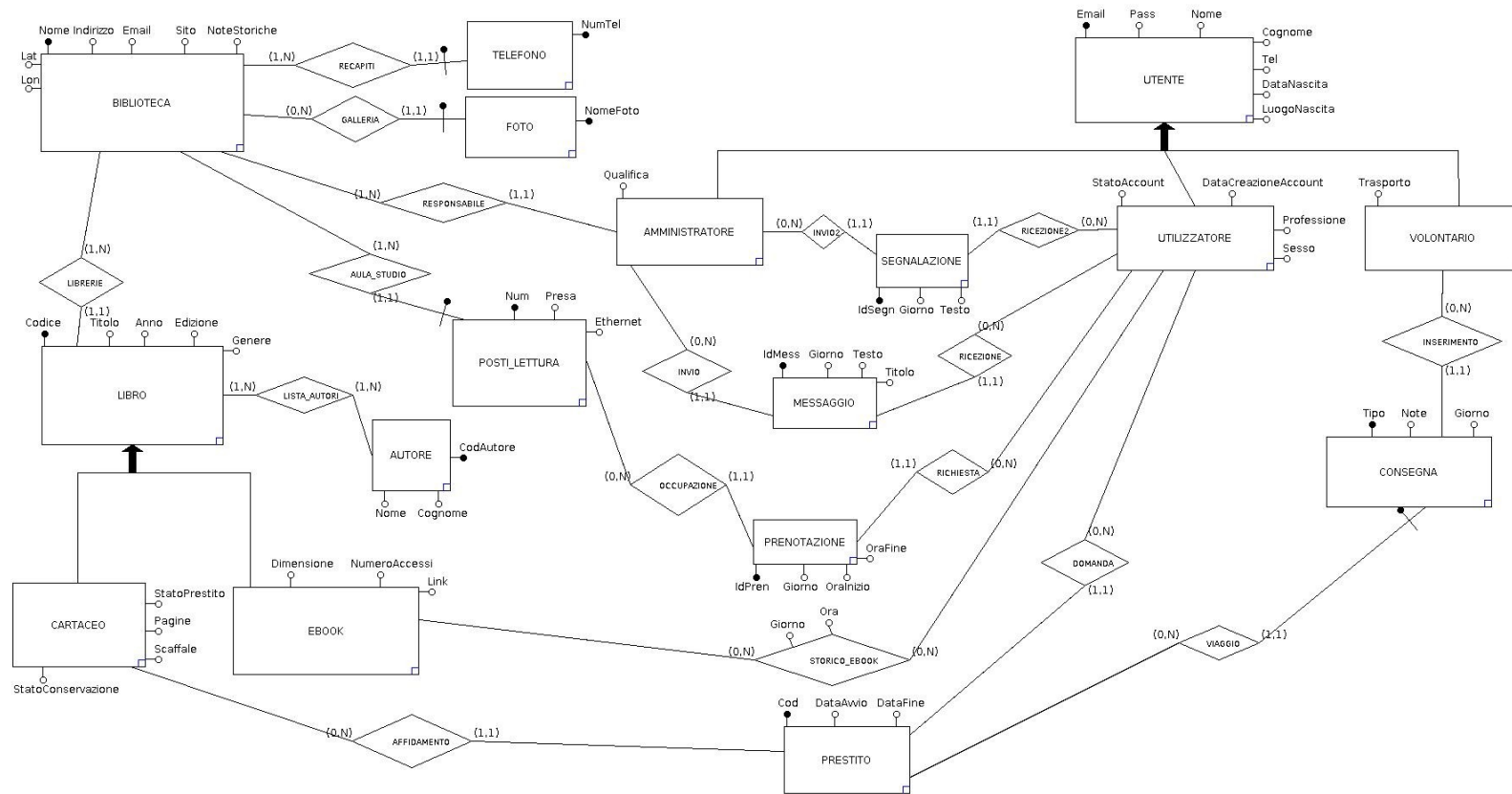


Figure 2: Diagramma E-R db.ebiblio

2.2 Dizionario delle entità/relazioni

Entità	Descrizione	Attributi	Identificatore
BIBLIOTECA	Luogo universitario	Nome, Indirizzo, Email, Sito, NoteStoriche, Lat, Lon	Nome
TELEFONO	Recapito/i telefonico biblioteca	NumTel	NumTel, BIBLIOTECA
FOTO	Foto biblioteca	NomeFoto	NomeFoto, BIBLIOTECA
LIBRO	Libri in possesso delle biblioteche	Codice, Titolo, Anno, Edizione, Genere	Codice
LIBRO CARTACEO	Libri cartacei in possesso delle biblioteche	StatoPrestito, Pagine, Scaffale, StatoConservazione	LIBRO
LIBRO EBOOK	E-book in possesso delle biblioteche	Dimensione, Link, NumeroAccessi	LIBRO
AUTORE	Autore di uno o più libri/edizioni	Nome, Cognome, CodAutore	CodAutore
POSTILETTURA	Postazioni di studio nelle biblioteche	Num, Presa, Ethernet	Num, BIBLIOTECA
UTENTE	Utente iscritto alla piattaforma	Email, Pass, Nome, Cognome, Tel, DataNascita, LuogoNascita	Email
UTENTE AMMINISTRATORE	Amministratore della piattaforma	Qualifica	UTENTE
UTENTE UTILIZZATORE	Utilizzatore della piattaforma	StatoAccount, DataCreazioneAccount, Professione, Sesso	UTENTE
UTENTE VOLONTARIO	Volontario consegne libri	Trasporto	UTENTE
MESSAGGIO	Messaggi mandati dagli amministratori agli utenti	IdMess, Giorno, Testo, Titolo	IdMess
SEGNALAZIONE	Segnalazioni inviate dagli amministratori agli utenti	IdSegn, Giorno, Testo	IdSegn
PRENOTAZIONE	Prenotazioni dei posti nelle aule studio	IdPren, Giorno, OraInizio, OraFine	IdPren
CONSEGNA	Consegna/Restituzione dei libri cartacei presi in prestito	Tipo, Note, Giorno	PRESTITO
PRESTITO	Prestiti dei libri cartacei	Cod, DataAvvio, DataFine	Cod

Table 3: Dizionario delle entità db.ebiblio

Relazione	Descrizione	Tipo	Componenti	Attributi
RECAPITI	Associa ad una libreria il/i suo/-suoi recapiti telefonici	Uno-a-molti	TELEFONO, BIBLIOTECA	
GALLERIA	Associa ad una libreria le sue foto	Uno-a-molti	FOTO, BIBLIOTECA	
RESPONSABILE	Associa ad ogni biblioteca uno o più amministratori	Uno-a-molti	UTENTE AMMINISTRATORE, BIBLIOTECA	
AULA_STUDIO	Associa ad ogni biblioteca i suoi posti lettura	Uno-a-molti	POSTI LETTURA, BIBLIOTECA	
LIBRERIE	Associa ad ogni biblioteca i libri che sono conservati in essa	Uno-a-molti	LIBRO, BIBLIOTECA	
LISTA_AUTORI	Associa ogni libro alla lista di suoi autori, ovviamente un autore può aver scritto più libri o edizioni diverse	Molti-a-molti	LIBRO, AUTORE	
INSERIMENTO	Associa un volontario alla consegna dei libri richiesti in prestito	Uno-a-molti	CONSEGNA, UTENTE VOLONTARIO	
VIAGGIO	Associa ogni prestito alle sue consegne di affidamento e restituzione	Uno-a-molti	CONSEGNA, PRESTITO	
AFFIDAMENTO	Associa ogni prestito al libro richiesto	Uno-a-molti	PRESTITO, LIBRO CARTACEO	
DOMANDA	Associa ogni domanda di prestito ad un utente utilizzatore della piattaforma	Uno-a-molti	PRESTITO, UTENTE UTILIZZATORE	
STORICO_EBOOK	Associa gli accessi degli utilizzatori alle schede degli e-book	Molti-a-molti	UTENTE UTILIZZATORE, LIBRO EBOOK	Giorno, Ora
OCCUPAZIONE	Associa ogni prenotazione al posto richiesto in aula studio	Uno-a-molti	PRENOTAZIONE, POSTI LETTURA	
RICHIESTA	Associa ogni prenotazione a chi ha effettuato la richiesta	Uno-a-molti	PRENOTAZIONE, UTENTE UTILIZZATORE	
INVIO	Associa ogni messaggio all'amministratore che l'ha mandato	Uno-a-molti	MESSAGGIO, UTENTE AMMINISTRATORE	
RICEZIONE	Associa ogni messaggio all'utente utilizzatore che l'ha ricevuto	Uno-a-molti	MESSAGGIO, UTENTE UTILIZZATORE	
INVIO2	Associa ogni segnalazione all'amministratore che l'ha mandata	Uno-a-molti	SEGNALAZIONE, UTENTE AMMINISTRATORE	
RICEZIONE2	Associa ogni segnalazione all'utente che l'ha ricevuta	Uno-a-molti	SEGNALAZIONE, UTENTE UTILIZZATORE	

Table 4: Dizionario delle relazioni db.ebiblio

2.3 Tavola delle business rules

Regole di vincolo:

- POSTO_LETTURA.Presa è un campo booleano.
- POSTO_LETTURA.Ethernet è un campo booleano.
- CARTACEO.StatoConservazione può essere solo: Ottimo, Buono, Non Buono, Scadente.
- CARTACEO.StatoPrestito può essere solo: Disponibile, Prenotato o Consegnato.
- UTILIZZATORE.StatoAccount può essere solo: Attivo o Sospeso.
- CONSEGNA.Tipo può essere solo: Restituzione o Affidamento.
- CONSEGNA.Note è un campo testuale di massimo 200 caratteri.
- AMMINISTRATORE.Qualifica è un campo testuale di massimo 10 caratteri.

Regole di derivazione:

- PRESTITO.DataFine = CONSEGNA.Giorno + 15 giorni
- Quando un utilizzatore riceve più di tre segnalazioni (anche da diverse biblioteche) lo stato dell'account viene settata a "Sospeso" e viene impedito qualsiasi accesso alla piattaforma all'utente.

3 Progettazione Logica

3.1 Ristrutturazione dello schema concettuale

Eliminazione delle generalizzazioni.

Per ridurre i valori null, dato che si tratta di due generalizzazioni totali, abbiamo scelto le seguenti soluzioni:

- Per la generalizzazione che coinvolge le entità UTENTE, AMMINISTRATORE, VOLONTARIO e UTILIZZATORE abbiamo usato la soluzione 2, accorpate l'entità genitore in quelle figlie.
- Per la generalizzazione che coinvolge le entità LIBRO, CARTACEO e EBOOK abbiamo usato la soluzione 3, sostituire le generalizzazioni con relazioni tra l'entità genitore e quelle figlie; quest'ultima scelta è stata preferita alla soluzione 2 per garantire l'univocità dell'attributo Codice per i libri (siano essi cartacei che ebook).

3.2 Analisi delle ridondanze

Analisi della ridondanza del campo NUMERO_ACCESSI per ogni ebook.

Operazioni:

1. Inserire un nuovo e-book (3 volte/mese, interattiva)
2. Rimuovere un e-book (3 volte/mese, interattiva)
3. Contare il numero di accessi di ogni e-book (2 volte/mese, interattiva)

Costanti:

- Peso operazioni scrittura $a = 2$
- Peso operazioni interattive $w_I = 1$

Tavola dei volumi:

- 50 accessi alla scheda di ogni e-book conservati nello STORICO_EBOOK
- 100 ebook totali

In generale per tutti i calcoli di costo operativo:

$$C(\text{Operazione}) = \text{frequenza_operazione} * \text{coeff_tipo_di_operazione} * (\text{peso_operazioni_scrittura} * \text{num_scritture_del_modello_E} - R + \text{num_totale_letture_del_modello_E} - R)$$

Costo delle operazioni senza NUMERO_ACCESSI:

1. Operazioni che vengono effettuate:
 - Scrivere una nuova riga per inserire un nuovo e-book
 - Scrivere una nuova coppia $\langle \text{Autore}, \text{Ebook} \rangle$

$$C(O_1) = 3 * 1 * (2 * 2 + 0) = 12$$

2. Operazioni che vengono effettuate:

- Rimuovere un e-book (scrivere un null nella riga corrispondente)
- Rimuovere cinquanta righe dell'accesso alla scheda dell'e-book dallo storico

$$C(O_2) = 3 * 1 * (2 * 50 + 0) = 66$$

3. Operazioni che vengono effettuate:

- Leggere 100 righe degli e-book
- Leggere 50 righe per ogni e-book dallo storico e-book (50*100 letture)

$$C(O_3) = 2 * 1 * (2 * 0 + (100 + 50 * 100)) = 10200$$

Costo delle operazioni con NUMERO_ACCESSI

L'unica operazione il cui costo verrebbe cambiato in caso di introduzione dell'attributo NUMERO_ACCESSI associato all'entità EBOOK è l'operazione 3 che diventerebbe:

- Leggere l'attributo NUMERO_ACCESSI delle 100 righe di EBOOK

$$C(O_{3R}) = 2 * 1 * (2 * 0 + 100) = 200$$

Calcolo il costo totale delle operazioni in caso di ridondanza e non, ed effettuo un confronto sullo speed up delle operazioni

$$C(S) = C(O_1) + C(O_2) + C(O_3) = 12 + 66 + 10200 = 10278$$

$$C(S_R) = C(O_1) + C(O_2) + C(O_{3R}) = 12 + 66 + 200 = 278$$

$$S/S_R = 10278/278 = 36.97$$

La ridondanza migliora lo speedup delle operazioni di un fattore ~ 36.97

Calcolo l'occupazione di memoria aggiuntiva in caso di ridondanza ipotizzando

$$m(S) = X_{byte},$$

allora $m(S_R) = X_{byte} + \text{volume dell'entità EBOOK} * \text{dimensione del campo aggiuntivo} =$
 $= X_{byte} + 100 * 4_{byte} \text{ (perché si tratta di un numero)} = X + 400_{byte}$

Quindi per questa ridondanza sono richiesti 400 byte in più, a fronte di un fattore di speedup delle operazioni ~ 36.97 , quindi la ridondanza va MANTENUTA perché si ha una forte accelerazione della velocità di operazione a fronte di soli 400 byte di memoria occupata in più.

3.3 Lista delle tabelle

- BIBLIOTECA(Nome, Indirizzo, Email, Sito, NoteStoriche, Lat, Lon)
- TELEFONO(NumTel, NomeBiblioteca)
- FOTO(NomeFoto, NomeBib)
- LIBRO(Codice, Titolo, Anno, Edizione, Biblioteca, Genere)
- CARTACEO(Codice, StatoPrestito, Pagine, Scaffale, StatoConservazione)
- EBOOK(Codice, Dimensione, NumeroAccessi, Link)
- AUTORE(CodAutore, Nome, Cognome)
- LISTA_AUTORI(CodiceAutore, CodiceLibro)
- STORICO_EBOOK(CodEbook, EmailUtente, Giorno, Ora)
- POSTILLETTURA(Num, NomeBiblioteca, Presa, Ethernet)
- AMMINISTRATORE(Email, Pass, Nome, Cognome, Tel, DataNascita, LuogoNascita, Qualifica, Responsabile)
- UTILIZZATORE(Email, Pass, Sesso, Nome, Cognome, Tel, DataNascita, LuogoNascita, StatoAccount, DataCreazioneAccount, Professione)
- VOLONTARIO(Email, Pass, Nome, Cognome, Tel, DataNascita, LuogoNascita, Trasporto)
- CONSEGNA(CodPrestito, Tipo, Note, Giorno, EmailVol)
- PRESTITO(Cod, DataAvvio, DataFine, CodLibro, EmailUtilizzatore)
- PRENOTAZIONE(IdPren, Giorno, OraInizio, OraFine, NumPosto, Biblioteca, EmailUtilizzatore)
- MESSAGGIO(IdMess, Giorno, Testo, Titolo, EmailAmm, EmailUti)
- SEGNALAZIONE(IdSegn, Giorno, Testo, EmailAmm, EmailUti)

3.4 Lista dei vincoli inter-relazionali

Vincoli di chiavi:

TELEFONO.NomeBiblioteca → BIBLIOTECA.Nome
FOTO.NomeBib → BIBLIOTECA.Nome
LISTA_AUTORI.CodiceLibro → LIBRO.Codice
LISTA_AUTORI.CodiceAutore → AUTORE.CodAutore
STORICO_EBOOK.CodEbook → EBOOK.Codice
STORICO_EBOOK.EmailUtente → UTILIZZATORE.Email
POSTILLETTURA.NomeBiblioteca → BIBLIOTECA.Nome
CONSEGNA.CodPrestito → PRESTITO.Cod
CONSEGNA.EmailVol → VOLONTARIO.Email
AMMINISTRATORE.Responsabile → BIBLIOTECA.Nome
LIBRO.Biblioteca → BIBLIOTECA.Nome
CARTACEO.Codice → LIBRO.Codice
EBOOK.Codice → LIBRO.Codice
PRESTITO.CodLibro → CARTACEO.Codice

PRESTITO.EmailUtilizzatore →UTILIZZATORE.Email
PRENOTAZIONE.NumPosto→POSTI_LETTURA.Num
PRENOTAZIONE.Biblioteca →BIBLIOTECA.Nome
PRENOTAZIONE.EmailUtilizzatore →UTILIZZATORE.Email
MESSAGGIO.EmailAmm→ AMMINISTRATORE.Email
MESSAGGIO.EmailUti→ UTILIZZATORE.Email
SEGNALAZIONE.EmailAmm→ AMMINISTRATORE.Email
SEGNALAZIONE.EmailUti→ UTILIZZATORE.Email

4 Descrizione (ad alto livello) delle funzionalità dell'applicazione web

La piattaforma Ebiblio gestisce l'intero ecosistema di biblioteche dell'università di Bologna.

Qualsiasi utente, anche se non registrato alla piattaforma può visualizzare le informazioni delle biblioteche: email, sito, via, numeri di telefono, galleria di foto, mappa, note storiche, libri cartacei e gli ebook della biblioteca.

Sempre senza essere registrati si può avere accesso alla pagina con i risultati del clustering e le classifiche della piattaforma.

Cliccando sul bottone "Signup" utenti utilizzatori e volontari possono registrarsi alla piattaforma, per accedervi poi usando il tasto "Login".

Gli utilizzatori della piattaforma possono accedere a qualsiasi biblioteca per prenotare un libro cartaceo, verificare la presenza di un posto lettura e prenotarlo, o accedere al pdf di un ebook.

Ogni utilizzatore ha la propria pagina personale, accessibile cliccando sulla propria mail posta a sinistra del tasto di "Logout", in questa sezione potrà vedere: i posti lettura prenotati, i libri prenotati, quelli consegnati, i messaggi e le segnalazioni ricevute.

Gli utenti volontari, si occupano delle consegne dei libri, accedono sempre tramite il tasto login e si possono registrare tramite il tasto signup.

Nella schermata di un utente volontario, quest'ultimo può effettuare una consegna a domicilio di un libro prenotato o effettuare una restituzione, e modificare data e note delle consegne.

Gli amministratori della piattaforma accedono tramite una route privata, quindi immettendo l'indirizzo <http://localhost:3000/loginAdmin> nella barra di ricerca dell'indirizzo si potrà accedere alla pagina di amministrazione della biblioteca in cui lavora l'amministratore.

In questa pagina gli utenti amministratori possono: creare e modificare libri ed ebook, vedere le prenotazioni dei posti lettura, mandare messaggi agli utenti, segnalarli o assolverli se sono stati sospesi (con 3 segnalazioni si viene sospesi).

Per una questione di sicurezza abbiamo deciso che un admin non può registrarsi tramite la piattaforma, ma deve essere inserito manualmente nel database.

Per l'implementazione del server, abbiamo utilizzato il framework "expressJS" basato sul linguaggio "nodeJS".

Per questione di sicurezza ogni operazione di scrittura sul database è stato utilizzato il metodo POST mentre per la lettura il metodo GET. Inoltre solamente l'indirizzo "http://localhost:3000" è abilitato ad effettuare chiamate al server grazie all'utilizzo della policy CORS.

Ogni chiamata al database è stata implementata utilizzando stored procedures.

Infine per tenere traccia dell'utente loggato abbiamo implementato dei cookie lato server.

Per il sito abbiamo utilizzato il framework ReactJS basato sempre su linguaggio "nodeJS" con l'aiuto della libreria css "TailwindCSS".

Per installare le dipendenze abbiamo usato "npm" che è un package manager.

La libreria "Leaflet" è stata usata per l'implementazione della mappa.

4.1 Istruzioni di avvio

4.1.1 Prerequisiti

- mysql and mongodb servers

- Node Package Manager (NMP)
- Yarn

Puoi scaricarlo da <https://nodejs.org/en/download/>

4.1.2 Setup

Vai sulla cartella "server" ed esegui:

```
$ npm install
```

Vai sulla cartella "website" ed esegui:

```
$ npm install
```

Assicurati di aver accesso sia il server mysql che quello mongodb.

Cambia la configurazione della connessione al server con i tuoi host, porte, nome utente e password, all'interno del file "server.js"

4.2 Avvio

- Per avviare il server vai nella directory "server" ed esegui:

```
$ node server.js
```

- Per avviare il sito vai nella directory "website" ed esegui:

```
$ yarn start
```

5 Clustering utenti utilizzatori

Abbiamo applicato un modello di clustering basato su algoritmo di K-Means, attraverso il quale si segmentano gli utenti utilizzatori, sulla base della loro professione, età, genere e numero di richieste di prestiti di libri cartacei effettuati.

La cluster/segmentation analysis è un insieme usate di tecniche per raggruppare oggetti in classi tra loro omogenee, ossia con caratteristiche simili.

Queste tecniche prendono in input un insieme di elementi da dividere in cluster e un numero di cluster.

In output, determinano gli insiemi di elementi che compongono ogni cluster.

Abbiamo utilizzato l'algoritmo di clusterizzazione k-means, che richiede l'indicazione a priori del numero di cluster. I dati da classificare sono attributi con valori reali, nel caso si trattasse di attributi testuali sarebbe necessaria una riconversione del dominio in valori reali.

Questo algoritmo iterativo si basa sul concetto di distanza tra elementi, per ogni cluster si definisce un centroide, ossia un punto (immaginario o reale) al centro di un cluster, e itera 3 passi:

1. Inizializzazione: si definiscono i parametri di input per eseguire l'algoritmo;
2. Assegnazione del cluster: ogni data points viene assegnato al cluster (o centroide) più vicino;
3. Aggiornamento della posizione del centroide: ricalcola il punto esatto del centroide e di conseguenza ne modifica la sua posizione.

Per utilizzare questo algoritmo caricato i dati del nostro database sul software Weka.

Abbiamo sostituito a WEKA il seguente arff file:

```
1 % 4 attributes
2 % 13 instances
3
4 @RELATION UTENTIUTILIZZATORI
5
6 @ATTRIBUTE Email STRING
7 @ATTRIBUTE Eta NUMERIC
8 @ATTRIBUTE Sesso {M, F}
9 @ATTRIBUTE NumPrenotazioni NUMERIC
10
11 @DATA
12 carla@gmail.com,26,F,0
13 franco@gmail.com,35,M,2
14 gino@gmail.com,35,M,2
15 giovanna@gmail.com,27,F,0
16 luigi@gmail.com,24,M,0
17 marco@gmail.com,53,M,2
18 matteo@gmail.com,23,M,0
19 mauro@gmail.com,30,M,0
20 melissa@gmail.com,28,F,1
21 michele@gmail.com,22,M,2
22 piero@gmail.com,25,M,1
23 tiziano@gmail.com,60,M,5
24 vanessa@gmail.com,29,F,2
```

datiCluster.arff

Abbiamo svolto tre analisi di clustering attraverso l'algoritmo K-Means, per individuare quale causasse l'errore (*Within cluster sum of squared errors*) minore:

- Con due cluster → Cluster 0, Cluster 1
- Con tre cluster → Cluster 0, Cluster 1, Cluster 2

- Con quattro cluster → Cluster 0, Cluster 1, Cluster 2, Cluster 3

In tutte le nostre analisi abbiamo ignorato l'attributo email perché univoco per ogni istanza e non significativo.

5.1 K-Means con due cluster

```

1  == Run information ==
2
3  Scheme:          weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-
                    pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -V -N 2 -A "weka.core.
                    EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
4  Relation:        UTENTIUTILIZZATORI
5  Instances:        13
6  Attributes:       4
7                    Eta
8                    Sesso
9                    NumPrenotazioni
10 Ignored:          Email
11
12 Test mode:         evaluate on training data
13
14
15 == Clustering model (full training set) ==
16
17
18
19 kMeans
20 ==
21
22 Number of iterations: 2
23 Within cluster sum of squared errors: 1.9617482302246845
24
25 Initial starting points (random):
26
27 Cluster 0: 35,M,2
28 Cluster 1: 26,F,0
29
30 Missing values globally replaced with mean/mode
31
32 Final cluster centroids:
33
34 Attribute          Full Data          Cluster#
35                   (13.0)          (9.0)          (4.0)
36
37 Eta                32.0769          34.1111          27.5
38                   +/-11.6438 +/-13.6971 +/-1.291
39
40 Sesso              M              M              F
41 M                  9.0 ( 69%) 9.0 (100%) 0.0 ( 0%)
42 F                  4.0 ( 30%) 0.0 ( 0%) 4.0 (100%)
43
44 NumPrenotazioni    1.3077          1.5556          0.75
45                   +/-1.4367 +/-1.5899 +/-0.9574
46
47
48
49
50
51 Time taken to build model (full training data) : 0 seconds
52
53 == Model and evaluation on training set ==
54
55 Clustered Instances
56

```

```

57 0      9 ( 69%)
58 1      4 ( 31%)

```

Risultati con due cluster

Abbiamo anche fatto mostrare media e deviazione standard per ogni attributo nei vari cluster. L'errore di questa analisi è ~ 1.96 .

Di seguito l'elenco degli utenti che appartengono a ciascun cluster

```

1 0 carla@gmail.com,26,F,0 cluster1
2 1 franco@gmail.com,35,M,2 cluster0
3 2 gino@gmail.com,35,M,2 cluster0
4 3 giovanna@gmail.com,27,F,0 cluster1
5 4 luigi@gmail.com,24,M,0 cluster0
6 5 marco@gmail.com,53,M,2 cluster0
7 6 matteo@gmail.com,23,M,0 cluster0
8 7 mauro@gmail.com,30,M,0 cluster0
9 8 melissa@gmail.com,28,F,1 cluster1
10 9 michele@gmail.com,22,M,2 cluster0
11 10 piero@gmail.com,25,M,1 cluster0
12 11 tiziano@gmail.com,60,M,5 cluster0
13 12 vanessa@gmail.com,29,F,2 cluster1

```

Assegnazioni con due cluster

5.2 K-Means con tre cluster

```

1 == Run information ==
2
3 Scheme:          weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-
                  pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -V -N 3 -A "weka.core.
                  EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
4 Relation:        UTENTIUTILIZZATORI
5 Instances:       13
6 Attributes:      4
7                  Eta
8                  Sesso
9                  NumPrenotazioni
10 Ignored:         Email
11 Test mode:       evaluate on training data
12
13
14
15 == Clustering model (full training set) ==
16
17 kMeans
18 =====
19
20
21 Number of iterations: 2
22 Within cluster sum of squared errors: 0.8754168975069254
23
24 Initial starting points (random):
25
26 Cluster 0: 35,M,2
27 Cluster 1: 26,F,0
28 Cluster 2: 25,M,1
29
30 Missing values globally replaced with mean/mode
31
32 Final cluster centroids:
33
34 Attribute          Full Data          Cluster#
35                                     0          1          2

```

```

35          (13.0)      (4.0)      (4.0)      (5.0)
36 =====
37 Eta          32.0769      45.75      27.5      24.8
38          +/-11.6438 +/-12.7377 +/-1.291 +/-3.1145
39
40 Sesso          M          M          F          M
41      M          9.0 ( 69%) 4.0 (100%) 0.0 ( 0%) 5.0 (100%)
42      F          4.0 ( 30%) 0.0 ( 0%) 4.0 (100%) 0.0 ( 0%)
43
44 NumPrenotazioni 1.3077      2.75      0.75      0.6
45          +/-1.4367      +/-1.5 +/-0.9574 +/-0.8944
46
47
48
49
50
51 Time taken to build model (full training data) : 0 seconds
52
53 == Model and evaluation on training set ==
54
55 Clustered Instances
56
57 0          4 ( 31%)
58 1          4 ( 31%)
59 2          5 ( 38%)

```

Risultati con tre cluster

L'errore di questa analisi è ~ 0.875 .

Di seguito l'elenco degli utenti che appartengono a ciascun cluster

```

1 0 carla@gmail.com,26,F,0 cluster1
2 1 franco@gmail.com,35,M,2 cluster0
3 2 gino@gmail.com,35,M,2 cluster0
4 3 giovanna@gmail.com,27,F,0 cluster1
5 4 luigi@gmail.com,24,M,0 cluster2
6 5 marco@gmail.com,53,M,2 cluster0
7 6 matteo@gmail.com,23,M,0 cluster2
8 7 mauro@gmail.com,30,M,0 cluster2
9 8 melissa@gmail.com,28,F,1 cluster1
10 9 michele@gmail.com,22,M,2 cluster2
11 10 piero@gmail.com,25,M,1 cluster2
12 11 tiziano@gmail.com,60,M,5 cluster0
13 12 vanessa@gmail.com,29,F,2 cluster1

```

Assegnazioni con tre cluster

5.3 K-Means con quattro cluster

```

1 == Run information ==
2
3 Scheme:          weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-
                  pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -V -N 4 -A "weka.core.
                  EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
4 Relation:        UTENTILUTILIZZATORI
5 Instances:        13
6 Attributes:       4
7                   Eta
8                   Sesso
9                   NumPrenotazioni
10 Ignored:
11                   Email
12 Test mode:        evaluate on training data

```

```

13
14
15 == Clustering model (full training set) ==
16
17
18 kMeans
19 ==
20
21 Number of iterations: 2
22 Within cluster sum of squared errors: 0.4179168975069252
23
24 Initial starting points (random):
25
26 Cluster 0: 35,M,2
27 Cluster 1: 26,F,0
28 Cluster 2: 25,M,1
29 Cluster 3: 60,M,5
30
31 Missing values globally replaced with mean/mode
32
33 Final cluster centroids:
34
35 Attribute          Full Data          Cluster#
36                   (13.0)          0          1          2          3
37                   (13.0)          (3.0)          (4.0)          (5.0)          (1.0)
38
39 Eta                  32.0769                  41          27.5          24.8          60
40                   +/-11.6438 +/-10.3923   +/-1.291   +/-3.1145   +/-NaN
41
42 Sesso                M                M                F                M                M
43   M                9.0 ( 69%) 3.0 (100%) 0.0 ( 0%) 5.0 (100%) 1.0 (100%)
44   F                4.0 ( 30%) 0.0 ( 0%) 4.0 (100%) 0.0 ( 0%) 0.0 ( 0%)
45
46 NumPrenotazioni      1.3077                2          0.75          0.6          5
47                   +/-1.4367                +/-0   +/-0.9574   +/-0.8944   +/-NaN
48
49
50
51
52 Time taken to build model (full training data) : 0 seconds
53
54 == Model and evaluation on training set ==
55
56 Clustered Instances
57
58 0          3 ( 23%)
59 1          4 ( 31%)
60 2          5 ( 38%)
61 3          1 ( 8%)

```

Risultati con quattro cluster

L'errore di questa analisi è ~ 0.417 .

Di seguito l'elenco degli utenti che appartengono a ciascun cluster

```

1 0 carla@gmail.com,26,F,0 cluster1
2 1 franco@gmail.com,35,M,2 cluster0
3 2 gino@gmail.com,35,M,2 cluster0
4 3 giovanna@gmail.com,27,F,0 cluster1
5 4 luigi@gmail.com,24,M,0 cluster2
6 5 marco@gmail.com,53,M,2 cluster0
7 6 matteo@gmail.com,23,M,0 cluster2
8 7 mauro@gmail.com,30,M,0 cluster2
9 8 melissa@gmail.com,28,F,1 cluster1
10 9 michele@gmail.com,22,M,2 cluster2

```

11	10	piero@gmail.com,25,M,1	cluster2
12	11	tiziano@gmail.com,60,M,5	cluster3
13	12	vanessa@gmail.com,29,F,2	cluster1

Assegnazioni con quattro cluster

5.4 Conclusioni sulla clusterizzazione

Quindi, dato che la differenza degli errori tra l'uso di tre cluster e di quattro è molto evidente, secondo la nostra analisi sarebbe preferibile implementare un algoritmo di clustering K-Means con quattro cluster.

A Codice SQL completo dello schema della base di dati

A.1 Creazione delle tabelle

```
1  create table BIBLIOTECA(  
2  Nome varchar(40) PRIMARY KEY,  
3  Indirizzo varchar(50) NOT NULL,  
4  Email varchar(40) UNIQUE,  
5  Sito varchar(40) UNIQUE,  
6  NoteStoriche varchar(300),  
7  Lat float(11,7) NOT NULL,  
8  Lon float(11,7) NOT NULL  
9  ) ENGINE=InnoDB;  
10 create table TELEFONO(  
11 NumTel varchar(15),  
12 NomeBiblioteca varchar(40),  
13 PRIMARY KEY(NumTel, NomeBiblioteca),  
14 FOREIGN KEY (NomeBiblioteca) REFERENCES BIBLIOTECA(Nome)  
15                                     ON DELETE CASCADE  
16                                     ON UPDATE CASCADE  
17 ) ENGINE=InnoDB;  
18 create table FOTO(  
19 NomeFoto varchar(40),  
20 NomeBib varchar(40),  
21 PRIMARY KEY(NomeFoto, NomeBib),  
22 FOREIGN KEY (NomeBib) REFERENCES BIBLIOTECA(Nome)  
23                                     ON DELETE CASCADE  
24                                     ON UPDATE CASCADE  
25 ) ENGINE=InnoDB;  
26 create table LIBRO(  
27 Codice int AUTO_INCREMENT PRIMARY KEY,  
28 Titolo varchar(50),  
29 Anno smallint,  
30 Edizione varchar(30),  
31 Biblioteca varchar(40),  
32 Genere varchar(30),  
33 FOREIGN KEY (Biblioteca) REFERENCES BIBLIOTECA(Nome)  
34                                     ON DELETE CASCADE  
35                                     ON UPDATE CASCADE  
36 ) ENGINE=InnoDB;  
37 create table CARTACEO(  
38 Codice int PRIMARY KEY,  
39 StatoPrestito varchar(11) CHECK(StatoPrestito="Disponibile" OR  
40 StatoPrestito="Prenotato" OR StatoPrestito="Consegnato"),  
41 Pagine smallint,  
42 Scaffale smallint,  
43 StatoConservazione varchar(9) CHECK(StatoConservazione="Ottimo" OR  
44 StatoConservazione="Buono" OR StatoConservazione="Non Buono" OR  
45 StatoConservazione="Scadente"),  
46 FOREIGN KEY (Codice) REFERENCES LIBRO(Codice)  
                                     ON DELETE CASCADE  
                                     ON UPDATE CASCADE  
47 ) ENGINE=InnoDB;
```

```

47 create table EBOOK(
48 Codice int PRIMARY KEY ,
49 Dimensione varchar(10),
50 NumeroAccessi int DEFAULT 0,
51 Link varchar(2100),
52 FOREIGN KEY (Codice) REFERENCES LIBRO(Codice)
53                                     ON DELETE CASCADE
54                                     ON UPDATE CASCADE
55 ) ENGINE=InnoDB;
56 create table AUTORE(
57 CodAutore int AUTO_INCREMENT PRIMARY KEY,
58 Nome varchar(20) NOT NULL,
59 Cognome varchar(30) NOT NULL
60 ) ENGINE=InnoDB;
61 create table AMMINISTRATORE(
62 Email varchar(30) PRIMARY KEY,
63 Pass varchar(20) NOT NULL,
64 Nome varchar(20),
65 Cognome varchar(30),
66 Tel varchar(15),
67 DataNascita date,
68 LuogoNascita varchar(20),
69 Qualifica varchar(10),
70 Responsabile varchar(40),
71 FOREIGN KEY (Responsabile) REFERENCES BIBLIOTECA(Nome)
72                                     ON DELETE CASCADE
73                                     ON UPDATE CASCADE
74 ) ENGINE=InnoDB;
75 create table UTILIZZATORE(
76 Email varchar(30) PRIMARY KEY,
77 Pass varchar(20) NOT NULL,
78 Sesso char(1) CHECK(Sesso="M" OR Sesso="F"),
79 Nome varchar(20),
80 Cognome varchar(30),
81 Tel varchar(15),
82 DataNascita date,
83 LuogoNascita varchar(20),
84 StatoAccount varchar(7) DEFAULT "Attivo" CHECK(StatoAccount="Attivo" OR
85 StatoAccount="Sospeso"),
86 DataCreazioneAccount date,
87 Professione varchar(20)
88 ) ENGINE=InnoDB;
89 create table VOLONTARIO(
90 Email varchar(30) PRIMARY KEY,
91 Pass varchar(20) NOT NULL,
92 Nome varchar(20),
93 Cognome varchar(30),
94 Tel varchar(15),
95 DataNascita date,
96 LuogoNascita varchar(20),
97 Trasporto varchar(15)
98 ) ENGINE=InnoDB;

```

```

98  create table LISTA_AUTORI(
99  CodiceAutore int,
100  CodiceLibro int,
101  FOREIGN KEY (CodiceAutore) REFERENCES AUTORE(CodAutore)
102
103
104  FOREIGN KEY (CodiceLibro) REFERENCES LIBRO(Codice)
105
106
107  PRIMARY KEY(CodiceAutore, CodiceLibro)
108  ) ENGINE=InnoDB;
109  create table STORICO_EBOOK(
110  CodEbook int,
111  EmailUtente varchar(30),
112  Giorno date NOT NULL,
113  Ora time NOT NULL,
114  PRIMARY KEY(CodEbook, EmailUtente, Giorno, Ora),
115  FOREIGN KEY (CodEbook) REFERENCES EBOOK(Codice)
116
117
118  FOREIGN KEY (EmailUtente) REFERENCES UTILIZZATORE(Email)
119
120
121  ) ENGINE=InnoDB;
122  create table POSTI_LETTURA(
123  Num int NOT NULL,
124  NomeBiblioteca varchar(40),
125  Presa boolean,
126  Ethernet boolean,
127  PRIMARY KEY(Num, NomeBiblioteca),
128  FOREIGN KEY (NomeBiblioteca) REFERENCES BIBLIOTECA(Nome)
129
130
131  ) ENGINE=InnoDB;
132  create table PRESTITO(
133  Cod int AUTO_INCREMENT PRIMARY KEY,
134  DataAvvio date,
135  DataFine date,
136  CodLibro int,
137  EmailUtilizzatore varchar(30),
138  FOREIGN KEY (CodLibro) REFERENCES CARTACEO(Codice)
139
140
141  FOREIGN KEY (EmailUtilizzatore) REFERENCES UTILIZZATORE(Email)
142
143
144  ) ENGINE=InnoDB;
145  create table CONSEGNA(
146  CodPrestito int,
147  Tipo varchar(12) CHECK(Tipo="Restituzione" OR Tipo="Affidamento"),

```

```

ON DELETE CASCADE
ON UPDATE
CASCADE,

```

```

ON DELETE CASCADE
ON UPDATE
CASCADE,

```

```

ON DELETE CASCADE
ON UPDATE CASCADE,

```

```

ON DELETE CASCADE
ON UPDATE CASCADE

```

```

ON DELETE CASCADE
ON UPDATE CASCADE

```

```

ON DELETE CASCADE
ON UPDATE CASCADE,

```

```

ON DELETE CASCADE
ON UPDATE CASCADE

```

```

148 Note varchar(200),
149 Giorno date NOT NULL,
150 EmailVol varchar(30),
151 PRIMARY KEY(CodPrestito, Tipo),
152 FOREIGN KEY (CodPrestito) REFERENCES PRESTITO(Cod)
153
154
155 FOREIGN KEY (EmailVol) REFERENCES VOLONTARIO(Email)
156
157
158 ) ENGINE=InnoDB;
159 create table PRENOTAZIONE(
160 IdPren int AUTO_INCREMENT PRIMARY KEY,
161 Giorno date,
162 OraInizio time,
163 OraFine time,
164 NumPosto int,
165 Biblioteca varchar(40),
166 EmailUtilizzatore varchar(30),
167 FOREIGN KEY (NumPosto) REFERENCES POSTI_LETTURA(Num)
168
169
170 FOREIGN KEY (Biblioteca) REFERENCES BIBLIOTECA(Nome)
171
172
173 FOREIGN KEY (EmailUtilizzatore) REFERENCES UTILIZZATORE(Email)
174
175
176 ) ENGINE=InnoDB;
177 create table MESSAGGIO(
178 IdMess int AUTO_INCREMENT PRIMARY KEY,
179 Giorno date,
180 Testo varchar(300),
181 Titolo varchar(30),
182 EmailAmm varchar(30),
183 EmailUti varchar(30),
184 FOREIGN KEY (EmailAmm) REFERENCES AMMINISTRATORE(Email)
185
186
187 FOREIGN KEY (EmailUti) REFERENCES UTILIZZATORE(Email)
188
189
190 ) ENGINE=InnoDB;
191 create table SEGNALAZIONE(
192 IdSegn int AUTO_INCREMENT PRIMARY KEY,
193 Giorno date,
194 Testo varchar(300),
195 EmailAmm varchar(30),

```

ON DELETE CASCADE
ON UPDATE
CASCADE,

ON DELETE CASCADE
ON UPDATE CASCADE

ON DELETE CASCADE
ON UPDATE
CASCADE,

ON DELETE CASCADE
ON UPDATE
CASCADE,

ON DELETE CASCADE
ON UPDATE
CASCADE,

ON DELETE CASCADE
ON UPDATE
CASCADE,

ON DELETE CASCADE
ON UPDATE
CASCADE,

ON DELETE CASCADE
ON UPDATE
CASCADE,

```

196 EmailUti varchar(30),
197 FOREIGN KEY (EmailAmm) REFERENCES AMMINISTRATORE(Email)
198                                     ON DELETE CASCADE
199                                     ON UPDATE
                                     CASCADE,
200 FOREIGN KEY (EmailUti) REFERENCES UTILIZZATORE(Email)
201                                     ON DELETE CASCADE
202                                     ON UPDATE CASCADE
203 ) ENGINE=InnoDB;

```

A.2 Operazioni che riguardano tutti gli utenti

```

1  ##TUTTI GLI UTENTI
2  # Autenticazione alla piattaforma
3  # Tipo 0 = Utilizzatore,                               Tipo 1 = Volontario
4  DELIMITER $$
5  CREATE PROCEDURE Autenticazione(IN tipo bool, IN mail varchar(30), IN psw
  varchar(20))
6  BEGIN
7      IF (tipo=0) THEN
8          SELECT StatoAccount as Stato
9          FROM UTILIZZATORE
10         WHERE (Email=mail AND Pass=psw);
11     ELSEIF (tipo=1) THEN
12         SELECT *
13         FROM VOLONTARIO
14         WHERE (Email=mail AND Pass=psw);
15     ELSE
16         SELECT *
17         FROM AMMINISTRATORE
18         WHERE (Email=mail AND Pass=psw);
19     END IF;
20 END $$
21 DELIMITER ;
22
23 # Visualizzazioni biblioteche presenti
24 DELIMITER $$
25 CREATE PROCEDURE VisualBiblioteche()
26 BEGIN
27     SELECT Nome, Indirizzo, NomeFoto
28     FROM BIBLIOTECA JOIN FOTO ON (Nome=NomeBib)
29     WHERE NomeFoto LIKE "%1.jpeg";
30 END $$
31 DELIMITER ;
32
33 # Visualizzazioni foto biblioteca
34 DELIMITER $$
35 CREATE PROCEDURE FotoBib(IN BibliotecaScelta varchar(40))
36 BEGIN
37     SELECT *
38     FROM FOTO
39     WHERE NomeBib = BibliotecaScelta;

```

```

40 END $$
41 DELIMITER ;
42
43 # Visualizzazione biblioteca
44 DELIMITER $$
45 CREATE PROCEDURE VisualBiblioteca(IN NomeBiblioteca varchar(40))
46 BEGIN
47     SELECT *
48     FROM BIBLIOTECA
49     WHERE Nome=NomeBiblioteca;
50 END $$
51 DELIMITER ;
52
53
54 # Visualizzazione dei posti lettura presenti in ogni biblioteca
55 DELIMITER $$
56 CREATE PROCEDURE VisualPosti()
57 BEGIN
58     SELECT * FROM POSTI_LETTURA;
59 END $$
60 DELIMITER ;
61
62 # Visualizzazione dei libri disponibili in tutte le biblioteche
63 DELIMITER $$
64 CREATE PROCEDURE VisualLibri()
65 BEGIN
66     SELECT * FROM LIBRO;
67 END $$
68 DELIMITER ;
69
70 #Visualizzazione dei cartacei di una biblioteca
71 DELIMITER $$
72 CREATE PROCEDURE VisualCartaceiBib(in BibliotecaScelta varchar(40))
73 BEGIN
74     SELECT *
75     FROM CARTACEO JOIN LIBRO ON (CARTACEO.Codice=LIBRO.Codice)
76     WHERE Biblioteca = BibliotecaScelta;
77 END $$
78 DELIMITER;
79
80
81 #Visualizzazione degli ebook di una biblioteca
82 DELIMITER $$
83 CREATE PROCEDURE VisualEbookBib(in BibliotecaScelta varchar(40))
84 BEGIN
85     SELECT EBOOK.*, Titolo, Edizione, Anno, Genere
86     FROM EBOOK JOIN LIBRO ON (EBOOK.Codice=LIBRO.Codice)
87     WHERE Biblioteca = BibliotecaScelta;
88 END $$
89 DELIMITER;
90

```

```

91  #Visual Autori per titolo
92  DELIMITER $$
93  CREATE PROCEDURE VisualAutori (IN Codice INT)
94  BEGIN
95      SELECT * FROM AUTORE WHERE CodAutore IN
96      (SELECT CodiceAutore FROM LISTA_AUTORI WHERE CodiceLibro=Codice);
97  END $$
98  DELIMITER;
99
100
101  # Visualizzazione scheda di un E-BOOK
102  DELIMITER $$
103  CREATE PROCEDURE VisualEbook(IN CodiceEbook int)
104  BEGIN
105      UPDATE EBOOK SET NumeroAccessi=NumeroAccessi+1 WHERE
106      (EBOOK.Codice=CodiceEbook);
107
108      SELECT EBOOK.Codice, Titolo, Anno, Edizione, Genere ,Biblioteca, Dimensione,
109      NumeroAccessi, Link
110      FROM EBOOK JOIN LIBRO ON (EBOOK.Codice=LIBRO.Codice)
111      WHERE EBOOK.Codice=CodiceEbook;
112
113      SELECT * FROM AUTORE WHERE CodAutore IN
114      (SELECT CodiceAutore FROM LISTA_AUTORI WHERE CodiceLibro=CodiceEbook);
115  END $$
116  DELIMITER ;
117
118  #Visualizzazione numeri di telefono biblioteche
119  DELIMITER $$
120  CREATE PROCEDURE NumeriTelefono (IN BibliotecaScelta varchar(40))
121  BEGIN
122      SELECT NumTel
123      FROM TELEFONO
124      WHERE NomeBiblioteca=BibliotecaScelta;
125  END $$
126  DELIMITER;
127
128  #Visualizzazione informazioni libri cartacei
129  DELIMITER $$
130  CREATE PROCEDURE CartaceoScelto (IN CartaceoS varchar(40))
131  BEGIN
132      SELECT *
133      FROM LIBRO JOIN CARTACEO ON (LIBRO.Codice=CARTACEO.Codice)
134      WHERE LIBRO.Codice=CartaceoS;
135  END $$
136  DELIMITER;

```

A.3 Operazioni che riguardano solo gli utenti UTILIZZATORI

```
1  ##SOLO UTILIZZATORI
2  # Registrazione alla utente piattaforma
3  DELIMITER $$
4  CREATE PROCEDURE RegistrazioneUtente( IN mail varchar(30), IN psw varchar(20), IN
SessoUt char(1), IN NomeUt varchar(20), IN CognomeUt varchar(30), IN TelUt
varchar(15), IN DataNascitaUt date, IN LuogoNascitaUt varchar(20), IN
ProfessioneUt varchar(20))
5  BEGIN
6      SET @DataOdierna=CURDATE();
7      INSERT INTO UTILIZZATORE (Email, Pass, Sesso ,Nome, Cognome, Tel,
DataNascita, LuogoNascita, DataCreazioneAccount, Professione)
8      VALUES(mail, psw, SessoUt, NomeUt, CognomeUt, TelUt, DataNascitaUt,
LuogoNascitaUt, @DataOdierna, ProfessioneUt);
9  END $$
10 DELIMITER ;
11
12
13 # Posti lettura disponibili
14 DELIMITER $$
15 CREATE PROCEDURE PostiDisponibili(IN Inizio time, IN Fine time, IN Biblio
varchar(40), IN GiornoPren date)
16 BEGIN
17     SELECT *
18     FROM POSTI_LETTURA
19     WHERE NomeBiblioteca=Biblio AND Num NOT IN
20         (SELECT NumPosto
21          FROM PRENOTAZIONE
22          WHERE Giorno=GiornoPren AND Biblioteca=Biblio AND ((Inizio
BETWEEN OraInizio AND OraFine) OR (Fine BETWEEN OraInizio AND
OraFine)));
23 END $$
24 DELIMITER ;
25
26
27
28 # Prenotazione di un posto lettura
29 DELIMITER $$
30 CREATE PROCEDURE PrenotazionePosto(IN GiornoPren date, IN InizioPren time, IN
FinePren time, IN Num int, IN Biblio varchar(40), IN EmailUt varchar(30))
31 BEGIN
32     INSERT INTO PRENOTAZIONE(NumPosto, Biblioteca, EmailUtilizzatore, Giorno,
OraInizio, OraFine)
33     VALUES(Num, Biblio, EmailUt, GiornoPren, InizioPren, FinePren);
34 END $$
35 DELIMITER ;
36
37 #Visualizzazione prenotazioni di posti lettura da parte di un utente
38 DELIMITER $$
39 CREATE PROCEDURE VisualPostiUt(IN EmailUti varchar(30))
40 BEGIN
```



```

41      SELECT Giorno, OraInizio, OraFine, Num, Biblioteca, Presa, Ethernet
42  FROM PRENOTAZIONE
43  JOIN POSTI_LETTURA AS P ON (NumPosto = P.Num ) AND (Biblioteca =
44      P.NomeBiblioteca)
45  WHERE EmailUtilizzatore=EmailUti;
46  END $$
47  DELIMITER ;
48
49
50  # Prestito di un libro cartaceo
51  DELIMITER $$
52  CREATE PROCEDURE PrestitoCartaceo( IN CodiceLibro int, IN EmailUti varchar(30))
53  BEGIN
54      INSERT INTO PRESTITO(CodLibro, EmailUtilizzatore)
55      VALUES(CodiceLibro, EmailUti);
56      #LO STATO DEL PRESTITO CAMBIA CON UN TRIGGER
57      #DATA INIZIO E FINE VENGONO INSERITI DA UN TRIGGER ALL INSERIMENTO DELLA
58      CONSEGNA DEL LIBRO RICHiesto
59  END $$
60  DELIMITER ;
61
62  #TRIGGER PRESTITO CARTACEO
63  CREATE TRIGGER PrestitoCartaceo
64  AFTER INSERT ON PRESTITO
65  FOR EACH ROW
66  UPDATE CARTACEO SET StatoPrestito="Prenotato" WHERE (Codice=NEW.CodLibro);
67
68
69  # Visualizzazione delle proprie prenotazioni
70  CREATE VIEW PRESTITI_UT(Prestito, DataAvvio, DataFine, CodLibro,
71  EmailUtilizzatore, Titolo)
72  AS SELECT Cod, DataAvvio, DataFine, CodLibro, EmailUtilizzatore, Titolo
73  FROM PRESTITO JOIN LIBRO ON CodLibro=Codice;
74  DELIMITER $$
75  CREATE PROCEDURE PrestitiUtente(IN EmailUti varchar(30))
76  BEGIN
77      SELECT *
78      FROM PRESTITI_UT
79      WHERE EmailUtilizzatore = EmailUti;
80  END $$
81  DELIMITER ;
82
83
84
85  # Visualizzazione propri eventi di consegna
86  CREATE VIEW CONSEGNE_UT(Prestito,Titolo, Tipo, Note, EmailVol, DataAvvio,
87  DataFine, CodLibro, EmailUtilizzatore, StatoPrestito)
88  AS SELECT CodPrestito, Titolo, Tipo, Note, EmailVol, DataAvvio, DataFine,
89  CodLibro, EmailUtilizzatore, StatoPrestito

```

```

88         FROM PRESTITO JOIN CARTACEO ON (PRESTITO.CodLibro=CARTACEO.Codice)
89         JOIN CONSEGNA ON
90             (PRESTITO.Cod=CONSEGNA.CodPrestito)
91             JOIN LIBRO ON
92                 (PRESTITO.CodLibro=LIBRO.Codice);
91
92 DELIMITER $$
93 CREATE PROCEDURE VisualConsegne(IN EmailUt varchar(30))
94 BEGIN
95     SELECT *
96     FROM CONSEGNE_UT
97     WHERE EmailUtilizzatore = EmailUt and Tipo="Affidamento";
98 END $$
99 DELIMITER ;
100
101
102 #inserimento dati nello storico ebook
103 DELIMITER $$
104 CREATE PROCEDURE StoricoEbook(IN CodiceEbook int, IN EmailUt varchar(30))
105 BEGIN
106     SET @GiornoUt = CURDATE();
107     SET @OraUt = CURTIME();
108     INSERT INTO STORICO_EBOOK(CodEbook, EmailUtente, Giorno, Ora)
109     VALUES (CodiceEbook, EmailUt, @GiornoUt, @OraUt);
110 END $$
111 DELIMITER ;
112
113
114
115 #Visualizzazione messaggi ricevuti
116 DELIMITER $$
117 CREATE PROCEDURE VisualMessaggi(IN EmailUt varchar(30))
118 BEGIN
119     SELECT *
120     FROM MESSAGGIO
121     WHERE EmailUti = EmailUt;
122 END $$
123 DELIMITER ;
124
125 #Visualizzazioni segnalazioni ricevute
126 DELIMITER $$
127 CREATE PROCEDURE VisualSegnalazioni(IN EmailUt varchar(30))
128 BEGIN
129     SELECT *
130     FROM SEGNALE
131     WHERE EmailUti = EmailUt;
132 END $$
133 DELIMITER ;

```

A.4 Operazioni che riguardano solo gli utenti VOLONTARI

```
1  ##SOLO VOLONTARI
2
3  #Registrazione volontario alla piattaforma
4  DELIMITER $$
5  CREATE PROCEDURE RegistrazioneVolontario( IN mail varchar(30), IN psw
6  varchar(20), IN NomeVol varchar(20), IN CognomeVol varchar(30), IN TelVol
7  varchar(15), IN DataNascitaVol date, IN LuogoNascitaVol varchar(20), IN
8  TrasportoVol varchar(20))
9  BEGIN
10     INSERT INTO VOLONTARIO(Email, Pass, Nome, Cognome, Tel, DataNascita,
11     LuogoNascita, Trasporto)
12     VALUES(mail, psw, NomeVol, CognomeVol, TelVol, DataNascitaVol,
13     LuogoNascitaVol, TrasportoVol);
14 END $$
15 DELIMITER ;
16
17 # Visualizzazione di tutte le prenotazioni inserite sulla piattaforma
18 CREATE VIEW PRESTITI_VOL(Prestito, DataAvvio, DataFine, CodLibro,
19 EmailUtilizzatore, StatoPrestito, Biblioteca)
20 AS SELECT Cod, DataAvvio, DataFine, CodLibro, EmailUtilizzatore, StatoPrestito,
21 Biblioteca
22 FROM PRESTITO JOIN CARTACEO ON
23 (PRESTITO.CodLibro=CARTACEO.Codice)
24 JOIN LIBRO ON
25 (PRESTITO.Cod=LIBRO.Codice)
26 WHERE StatoPrestito <> "Disponibile";
27
28 DELIMITER $$
29 CREATE PROCEDURE VisualPrenotazioniCartei()
30 BEGIN
31     SELECT PRESTITI_VOL.*, Note, Giorno AS GiornoConsegna
32     FROM PRESTITI_VOL
33     LEFT JOIN CONSEGNA ON (CodPrestito = Prestito)
34     WHERE CONSEGNA.Tipo="Affidamento" or CONSEGNA.Tipo IS NULL;
35 END $$
36 DELIMITER ;
37
38 # Inserimento di un nuovo evento di consegna
39 DELIMITER $$
40 CREATE PROCEDURE InsertConsegna (IN NumPrestito int, IN Tipo varchar(12), IN Note
41 varchar(200), IN EmailVolontario varchar(30))
42 BEGIN
43     SET @Giorno=CURDATE();
44     INSERT INTO CONSEGNA(CodPrestito, Tipo, Note, Giorno, EmailVol)
45     VALUES(NumPrestito, Tipo, Note, @Giorno, EmailVolontario);
46 END $$
47 DELIMITER ;
48
49 #TRIGGER CONSEGNA
```

```

41 DELIMITER |
42 CREATE TRIGGER ConsegnaCartaceo
43 AFTER INSERT ON CONSEGNA
44 FOR EACH ROW
45 BEGIN
46     IF (NEW.Tipo="Affidamento") THEN
47         #INSERISCI DATA INIZIO E FINE SU PRESTITO
48         #La data di fine prestito viene calcolata come 15 giorni dalla
49         data di consegna
50         UPDATE PRESTITO SET DataAvvio = NEW.Giorno WHERE
51         (Cod=NEW.CodPrestito);
52         UPDATE PRESTITI_VOL SET DataAvvio = NEW.Giorno WHERE
53         (PRESTITI_VOL.Prestito=NEW.CodPrestito);
54         UPDATE PRESTITO SET DataFine = DATE_ADD(NEW.Giorno, INTERVAL 15
55         DAY) WHERE (Cod=NEW.CodPrestito);
56         UPDATE PRESTITI_VOL SET DataFine = DATE_ADD(NEW.Giorno, INTERVAL 15 DAY)
57         WHERE (PRESTITI_VOL.Prestito=NEW.CodPrestito);
58     ELSE
59         #se riconsegno prima aggiorno la data
60         UPDATE PRESTITO SET DataFine = NEW.Giorno WHERE
61         (Cod=NEW.CodPrestito);
62         UPDATE PRESTITI_VOL SET DataFine=NEW.Giorno WHERE
63         (PRESTITI_VOL.Prestito=NEW.CodPrestito);
64     END IF;
65     #CAMBIO LO STATO IN CONSEGNATO O DISPONIBILE
66     SET @CodiceLibro = 0;
67     SELECT CodLibro INTO @CodiceLibro
68     FROM PRESTITO
69     WHERE Cod=NEW.CodPrestito;
70     IF (NEW.Tipo="Affidamento") THEN
71         UPDATE CARTACEO SET StatoPrestito="Consegnato" WHERE
72         (Codice=@CodiceLibro);
73         UPDATE PRESTITI_VOL SET StatoPrestito="Consegnato" WHERE
74         (PRESTITI_VOL.Prestito=NEW.CodPrestito);
75     ELSE
76         UPDATE CARTACEO SET StatoPrestito="Disponibile" WHERE
77         (Codice=@CodiceLibro);
78         UPDATE PRESTITI_VOL SET StatoPrestito="Disponibile" WHERE
79         (PRESTITI_VOL.Prestito=NEW.CodPrestito);
80     END IF;
81 END |
82 DELIMITER ;
83
84 # Aggiornamento di un evento di consegna
85 DELIMITER $$
86 CREATE PROCEDURE UpdateConsegna (IN NumPrestito int, IN TipoConsegna varchar(12),
87 IN NoteC varchar(200),IN GiornoC date,IN EmailC varchar(30))
88 BEGIN
89     IF (NoteC IS NOT NULL AND NoteC <> "" AND NoteC <> " ") THEN

```

```

81         UPDATE CONSEGNA
82         SET Note=NoteC
83         WHERE CodPrestito=NumPrestito AND Tipo=TipoConsegna;
84     END IF;
85     IF (GiornoC IS NOT NULL AND GiornoC <> 0000-00-00) THEN
86         UPDATE CONSEGNA
87         SET Giorno=GiornoC
88         WHERE CodPrestito=NumPrestito AND Tipo=TipoConsegna;
89     END IF;
90     IF (EmailC IS NOT NULL AND EmailC <> "" AND EmailC <> " ") THEN
91         UPDATE CONSEGNA
92         SET EmailVol=EmailC
93         WHERE CodPrestito=NumPrestito AND Tipo=TipoConsegna;
94     END IF;
95 END $$
96 DELIMITER ;
97
98 DELIMITER |
99 CREATE TRIGGER UpdateConsegnaCartaceo
100 AFTER UPDATE ON CONSEGNA
101 FOR EACH ROW
102 BEGIN
103     IF (NEW.Giorno IS NOT NULL AND NEW.Giorno <> 0000-00-00) THEN
104         IF (NEW.Tipo="Affidamento") THEN
105             #INSERISCI DATA INIZIO E FINE SU PRESTITO
106             #La data di fine prestito viene calcolata come 15 giorni
107             #dalla data di consegna
108             UPDATE PRESTITO SET DataAvvio = NEW.Giorno WHERE
109             (Cod=NEW.CodPrestito);
110             UPDATE PRESTITI_VOL SET DataAvvio = NEW.Giorno WHERE
111             (PRESTITI_VOL.Prestito=NEW.CodPrestito);
112             UPDATE PRESTITO SET DataFine = DATE_ADD(NEW.Giorno,
113             INTERVAL 15 DAY) WHERE (Cod=NEW.CodPrestito);
114             UPDATE PRESTITI_VOL SET DataFine = DATE_ADD(NEW.Giorno,
115             INTERVAL 15 DAY) WHERE
116             (PRESTITI_VOL.Prestito=NEW.CodPrestito);
117         END IF;
118         #CAMBIO LO STATO IN CONSEGNATO O DISPONIBILE
119         SET @CodiceLibro = 0;
120         SELECT CodLibro INTO @CodiceLibro
121         FROM PRESTITO
122         WHERE Cod=NEW.CodPrestito;
123         IF (NEW.Tipo="Affidamento") THEN
124             UPDATE CARTACEO SET StatoPrestito="Consegnato" WHERE
125             (Codice=@CodiceLibro);
126             UPDATE PRESTITI_VOL SET StatoPrestito="Consegnato" WHERE
127             (PRESTITI_VOL.Prestito=NEW.CodPrestito);
128         END IF;
129     END IF;
130 END |
131 DELIMITER ;

```

A.5 Operazioni che riguardano solo gli utenti AMMINISTRATORI

```
1  #SOLO AMMINISTRATORI
2
3  #Biblioteca amministratore
4  DELIMITER $$
5  CREATE PROCEDURE BibliotecaAmministratore(IN EmailAmministratore varchar(30))
6  BEGIN
7      SELECT Responsabile as NomeBiblioteca
8      FROM AMMINISTRATORE
9      WHERE Email=EmailAmministratore;
10 END $$
11 DELIMITER ;
12
13 #Inserimento degli autori di un libro e collegamento con il libro
14 DELIMITER $$
15 CREATE PROCEDURE InsertAutori(IN CodLibro int, IN Autori varchar(100))
16 BEGIN
17     SET @delimiterCount = LENGTH(Autori) - LENGTH(REPLACE(Autori, ',', ''));
18     SET @autori_sub = SUBSTRING_INDEX(Autori, ',', 1);
19     SET @loopCount = 1;
20
21     WHILE @loopCount <= @delimiterCount + 1 DO
22         SET @autore = SUBSTRING_INDEX(Autori, ',', 1);
23         SET @AutoreEsistente = (SELECT CodAutore FROM AUTORE WHERE Nome=
24             SUBSTRING_INDEX(@autore, ' ', 1) AND COGNOME =
25             SUBSTRING_INDEX(@autore, ' ', -1));
26         SET @CoppiaEsistente = (SELECT CodiceAutore FROM LISTA_AUTORI
27             WHERE CodiceLibro=CodLibro AND CodiceAutore=@AutoreEsistente );
28
29         IF (@AutoreEsistente IS NULL) THEN
30             /*Put an author in table AUTORE*/
31             INSERT INTO AUTORE(Nome,Cognome)
32             VALUES (SUBSTRING_INDEX(@autore, ' ', 1), SUBSTRING_INDEX(@autore, '
33             ', -1)) ;
34             SET @codice = (SELECT Count(*) FROM AUTORE);
35             /*Connect author with book*/
36             INSERT INTO LISTA_AUTORI(CodiceAutore, CodiceLibro)
37             VALUES (@codice, CodLibro);
38         ELSEIF (@CoppiaEsistente IS NULL) THEN
39             /*Connect EXISTING author with book*/
40             INSERT INTO LISTA_AUTORI(CodiceAutore, CodiceLibro)
41             VALUES (@AutoreEsistente, CodLibro);
42         END IF;
43
44         /* Remove last used id from input string */
45         SET Autori = REPLACE(Autori, CONCAT(@autore, ','), '');
46         SET @loopCount = @loopCount + 1;
47     END WHILE;
48 END $$
49 DELIMITER ;
50
```

```

47
48 # Inserimento di un libro presso la biblioteca gestita
49 DELIMITER $$
50 CREATE PROCEDURE InsertLibro(IN Titolo varchar(50), IN Anno smallint, IN Edizione
51 varchar(30), IN Biblioteca varchar(40), Genere varchar(30),
52 IN Tipo Bool,
53 IN StatoPrestito varchar(11), IN Pagine smallint, IN Scaffale smallint, IN
54 StatoConservazione varchar(9),
55 IN Dimensione varchar(10), IN Link varchar(2100), IN Autori varchar(100) )
56 # se Tipo=true allora è un cartaceo, se Tipo=false allora è un ebook
57 BEGIN
58     INSERT INTO LIBRO(Titolo, Anno, Edizione, Biblioteca, Genere)
59     VALUES(Titolo, Anno, Edizione, Biblioteca, Genere);
60     SET @UltimoCodice = LAST_INSERT_ID();
61     CALL InsertAutori(@UltimoCodice,Autori);
62     IF Tipo="1" THEN
63         INSERT INTO CARTACEO(Codice, StatoPrestito, Pagine, Scaffale,
64         StatoConservazione)
65         VALUES (@UltimoCodice, StatoPrestito, Pagine, Scaffale,
66         StatoConservazione);
67     ELSE
68         INSERT INTO EBOOK(Codice, Dimensione, Link)
69         VALUES(@UltimoCodice, Dimensione, Link);
70     END IF;
71 END $$
72 DELIMITER ;
73
74 #Cancellazione di un libro presso la biblioteca gestita
75 DELIMITER $$
76 CREATE PROCEDURE DeleteLibro(IN CodiceLibro int)
77 BEGIN
78     DELETE FROM LIBRO
79     WHERE Codice=CodiceLibro;
80 END $$
81 DELIMITER ;
82
83 #Aggiornamento di un libro presso la biblioteca gestita
84 DELIMITER $$
85 CREATE PROCEDURE UpdateLibro(IN CodiceLibro int, IN BibliotecaGestita varchar(40)
86 ,
87     IN TitoloU varchar(50), IN AnnoU smallint, IN
88     EdizioneU varchar(30), IN GenereU varchar(30),
89     IN StatoPrestitoU varchar(11), IN PagineU
90     smallint, IN ScaffaleU smallint, IN
91     StatoConservazioneU varchar(9),
92     IN DimensioneU varchar(10), IN NumeroAccessiU int, IN LinkU
93     varchar(2100), IN AutoriU varchar(100))
94 BEGIN
95     IF (TitoloU IS NOT NULL AND TitoloU <> "" AND TitoloU <> " ") THEN
96         UPDATE LIBRO

```

```

90         SET Titolo=TitoloU
91         WHERE Codice=CodiceLibro AND Biblioteca=BibliotecaGestita;
92     END IF;
93     IF (AnnoU IS NOT NULL AND AnnoU <> 0) THEN
94         UPDATE LIBRO
95         SET Anno=AnnoU
96         WHERE Codice=CodiceLibro AND Biblioteca=BibliotecaGestita;
97     END IF;
98     IF (EdizioneU IS NOT NULL AND EdizioneU <> "" AND EdizioneU <> " ") THEN
99         UPDATE LIBRO
100        SET Edizione=EdizioneU
101        WHERE Codice=CodiceLibro AND Biblioteca=BibliotecaGestita;
102    END IF;
103    IF (GenereU IS NOT NULL AND GenereU <> "" AND GenereU <> " ") THEN
104        UPDATE LIBRO
105        SET Genere=GenereU
106        WHERE Codice=CodiceLibro AND Biblioteca=BibliotecaGestita;
107    END IF;
108    IF (StatoPrestitoU IS NOT NULL AND StatoPrestitoU <> "" AND StatoPrestitoU <>
" ") THEN
109        UPDATE CARTACEO
110        SET StatoPrestito=StatoPrestitoU
111        WHERE Codice=CodiceLibro;
112    END IF;
113    IF (PagineU IS NOT NULL AND PagineU <> 0) THEN
114        UPDATE CARTACEO
115        SET Pagine=PagineU
116        WHERE Codice=CodiceLibro;
117    END IF;
118    IF (ScaffaleU IS NOT NULL AND ScaffaleU <> 0) THEN
119        UPDATE CARTACEO
120        SET Scaffale=ScaffaleU
121        WHERE Codice=CodiceLibro;
122    END IF;
123    IF (StatoConservazioneU IS NOT NULL AND StatoConservazioneU <> "" AND
StatoConservazioneU <> " ") THEN
124        UPDATE CARTACEO
125        SET StatoConservazione=StatoConservazioneU
126        WHERE Codice=CodiceLibro;
127    END IF;
128    IF (DimensioneU IS NOT NULL AND DimensioneU <> "" AND DimensioneU <> " ")
THEN
129        UPDATE EBOOK
130        SET Dimensione=DimensioneU
131        WHERE Codice=CodiceLibro;
132    END IF;
133    IF (NumeroAccessiU IS NOT NULL AND NumeroAccessiU <> 0) THEN
134        UPDATE EBOOK
135        SET NumeroAccessi=NumeroAccessiU
136        WHERE Codice=CodiceLibro;
137    END IF;
138    IF (LinkU IS NOT NULL AND LinkU <> "" AND LinkU <> " ") THEN

```



```

139         UPDATE EBOOK
140         SET Link=LinkU
141         WHERE Codice=CodiceLibro;
142     END IF;
143     IF (AutoriU IS NOT NULL AND AutoriU <> "" AND AutoriU <> " ") THEN
144         DELETE FROM LISTA_AUTORI WHERE
145             LISTA_AUTORI.CodiceLibro=CodiceLibro;
146     CALL InsertAutori(CodiceLibro,AutoriU);
147     END IF;
148 END $$
149 DELIMITER ;
150
151
152 # Visualizzazione di tutte le prenotazioni presso la biblioteca gestita
153 CREATE VIEW PRENOTAZIONI_AMM(Prenotazione, Giorno, OraInizio, OraFine, NumPosto,
154     Biblioteca, EmailUtilizzatore, Presa, Ethernet)
155 AS SELECT IdPren, Giorno, OraInizio, OraFine, NumPosto, Biblioteca,
156     EmailUtilizzatore, Presa, Ethernet
157 FROM PRENOTAZIONE JOIN POSTI_LETTURA ON (NumPosto=Num AND
158     Biblioteca=NomeBiblioteca);
159
160 DELIMITER $$
161 CREATE PROCEDURE VisualPrenotazioniPosti(IN BibliotecaG varchar(40))
162 BEGIN
163     SELECT *
164     FROM PRENOTAZIONI_AMM
165     JOIN POSTI_LETTURA AS P ON (NumPosto = P.Num ) AND (Biblioteca =
166         P.NomeBiblioteca)
167     WHERE Biblioteca=BibliotecaG;
168 END $$
169 DELIMITER ;
170
171 # Inserimento di un messaggio rivolto ad un utente utilizzatore
172 DELIMITER $$
173 CREATE PROCEDURE InsertMessaggio(IN Titolo varchar(30), IN Testo varchar(300) ,
174     IN EmailAmministratore varchar(30), IN EmailUtilizzatore varchar(30))
175 BEGIN
176     INSERT INTO MESSAGGIO(Giorno, Testo, Titolo, EmailAmm, EmailUti)
177     VALUES (CURDATE(),Testo,Titolo,EmailAmministratore,EmailUtilizzatore);
178 END $$
179 DELIMITER ;
180
181 # Inserimento di una segnalazione di comportamento non corretto
182 DELIMITER $$
183 CREATE PROCEDURE InsertSegnalazione(IN Testo varchar(300), IN EmailAmministratore
184     varchar(30), IN EmailUtilizzatore varchar(30))
185 BEGIN
186     INSERT INTO SEGNALAZIONE(Giorno, Testo, EmailAmm, EmailUti)
187     VALUES (CURDATE(),Testo,EmailAmministratore,EmailUtilizzatore);
188 END $$

```

```

184 DELIMITER ;
185
186 #TRIGGER SOSPENSIONE
187 #Quando un utilizzatore riceve più di 3 segnalazioni il suo account viene settato
a "Sospeso"
188 DELIMITER |
189 CREATE TRIGGER Sospensione
190 AFTER INSERT ON SEGNALAZIONE
191 FOR EACH ROW
192 BEGIN
193 SET @CountSegnalazioni = 0;
194 SELECT Count(*) INTO @CountSegnalazioni
195 FROM SEGNALAZIONE
196 WHERE EmailUti = NEW.EmailUti;
197 IF (3 <= @CountSegnalazioni) THEN
198 UPDATE UTILIZZATORE
199 SET StatoAccount="Sospeso"
200 WHERE Email=NEW.EmailUti;
201 END IF;
202 END |
203 DELIMITER ;
204
205
206 # Assoluzione: Rimuovere tutte le segnalazioni di un utente, riportandone lo
stato ad Attivo
207 DELIMITER $$
208 CREATE PROCEDURE AssoluzioneUtente(IN EmailUtilizzatore varchar(30))
209 BEGIN
210 DELETE FROM SEGNALAZIONE WHERE EmailUti=EmailUtilizzatore;
211 UPDATE UTILIZZATORE
212 SET StatoAccount="Attivo"
213 WHERE Email=EmailUtilizzatore;
214 END $$
215 DELIMITER ;

```

A.6 Classifiche

```

1 #STATISTICHE
2
3 # Visualizzare la classifica dei volontari che hanno effettuato più consegne
4 DELIMITER $$
5 CREATE PROCEDURE ClassificaVol()
6 BEGIN
7 SELECT Count(EmailVol) AS "Number of deliveries", EmailVol as "Email
  Volunteer"
8 FROM CONSEGNA
9 GROUP BY EmailVol
10 ORDER BY Count(EmailVol) DESC;
11 END $$
12 DELIMITER ;
13
14 # Visualizzare la classifica dei libri cartacei più prenotati

```

```

15 DELIMITER $$
16 CREATE PROCEDURE ClassificaCartacei()
17 BEGIN
18     SELECT Count(CodLibro) as "Number of lending", CodLibro AS "Book code",
19           Titolo as "Title"
20     FROM PRESTITI_UT
21     GROUP BY CodLibro, Titolo
22     ORDER BY Count(CodLibro) DESC;
23 END $$
24 DELIMITER ;
25
26 # Visualizzare la classifica degli e-book più acceduti
27 DELIMITER $$
28 CREATE PROCEDURE ClassificaEbook()
29 BEGIN
30     SELECT Titolo as "Title", EBOOK.Codice as "E-book code", NumeroAccessi AS
31           "Number of accesses", Anno as "Year", Edizione as "Edition", Biblioteca
32           as "Library"
33     FROM EBOOK JOIN LIBRO ON (EBOOK.Codice=LIBRO.Codice)
34     ORDER BY NumeroAccessi DESC;
35 END $$
36 DELIMITER ;
37
38 # Visualizzare la classifica delle biblioteche con postazioni letture meno
39   utilizzate (in percentuale rispetto al numero di posti letture disponibili)
40 DELIMITER $$
41 CREATE PROCEDURE ClassificaBibliotecheMenoUsate()
42 BEGIN
43     SELECT Biblioteca as "Library" , Count(Distinct(IdPren)) as "Total
44           number of bookings", Count(Distinct(Num)) as "Number of seats",
45           CONCAT(
46                 TRUNCATE(((Count(Distinct(IdPren))/Count(Distinct(Num)))*100)
47                 ,2), "%") as "Percentage of occupation"
48     FROM PRENOTAZIONE JOIN POSTI_LETTURA ON (Biblioteca=NomeBiblioteca)
49     GROUP BY Biblioteca
50     ORDER BY (Count(Distinct(IdPren)))/(Count(Distinct(Num))) ASC;
51 END $$
52 DELIMITER ;

```