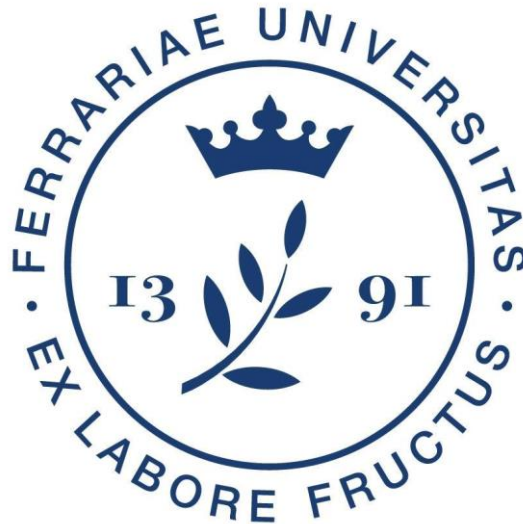


Università degli Studi di Ferrara

Gestione Biblioteche del Territorio

Progetto per il corso di Basi di Dati



Christian Bui – 144060

Marco Silvestrin – 150718

Anno accademico 2019/2020

Sommario

Capitolo 1 - Abstract	3
Capitolo 2 - Progettazione	4
2.1 Descrizione testuale delle entità.....	4
2.1.1 Biblioteca	4
2.1.2 Libro	4
2.1.3 Autore	4
2.1.4 Utente	5
2.1.5 Editore	5
2.1.6 Copie libro	5
2.1.7 Scritto da.....	6
2.1.8 Prestito	6
2.2 Descrizione testuale delle relazioni tra entità	7
2.2.1 Biblioteca – Libro: “Copie_libro”	7
2.2.1 Autore – Libro: “Scritto_da”	7
2.2.1 Utente – Copie_libro: “Prestito”	7
2.2.1 Editore – Libro	7
2.3 Modello E/R	8
2.4 Schema Relazionale	9
2.5 Normalizzazione Schema Relazionale	10
2.5.1 Prima Forma Normale (1NF)	10
2.5.2 Seconda e Terza Forma Normale (2NF/3NF)	10
Capitolo 3 - Costruzione.....	12
3.1 Introduzione.....	12
3.2 Interrogazioni della base di dati	12
3.2.1 Statistiche di base in algebra relazionale	12
Determinare le 5 lingue più comuni in cui sono scritti i libri presenti nella basi di dati.	14
3.2.2 Interrogazioni aggiuntive	14
Capitolo 4 - Realizzazione.....	16
4.1 HTML/CSS/PHP.....	16
4.1.1 HTML.....	16
4.1.2 CSS.....	16
4.1.3 PHP.....	17
4.2 Accessibilità.....	18
4.2.1 Barra di ricerca	18
4.2.2 Struttura e comportamento.....	19

Capitolo 1

Abstract

Il sito realizzato permette una gestione funzionale del sistema bibliotecario universitario di Ferrara. Si ha la necessità di avere una panoramica completa della gestione prestiti e utenti, compreso inserimento, modifica e cancellazione per entrambe le due entità. Ci si è posto innanzitutto davanti il problema della gestione dei libri e delle diverse copie, con necessità quindi di catalogare un libro non solo con i suoi dati basilari (quindi ISBN, titolo, ecc..) ma anche con un identificatore univoco per ogni singola copia di ogni libro, in modo da distinguere perfettamente a quale effettivamente ci stiamo riferendo. Inoltre, una volta capito il problema, era necessario interrogare la nostra base di dati per verificarne il corretto funzionamento con opportune statistiche che vedremo in seguito. Il sito web finale quindi si interfaccia con un ipotetico amministratore del sistema bibliotecario di Ferrara avente già i diritti di accesso (per questo motivo è sprovvisto di una schermata di login), che avrà a disposizione tutti gli elementi suddetti ed anche alcune funzionalità ulteriori per permettere una più agevole ed intuitiva navigazione.

Capitolo 2

Progettazione

2.1 Descrizione testuale delle entità

2.1.1 Biblioteca

L'entità biblioteca rappresenta tutte le informazioni relative alla sede specifica

Attributi

- Cod_biblioteca : intero <primary key>
- Indirizzo_bibl: *stringa*
- Nome_bibl: *stringa*
- Direttore_bibl: *stringa*

2.1.2 Libro

L'entità Libro contiene tutte le informazioni per ogni singolo libro

Attributi

- ISBN: *stringa* <primary key>
- Titolo_libro: *stringa*
- Anno_produzione: *stringa*
- Lingua: *stringa*
- Codice_editore intero

2.1.3 Autore

L'entità autore contiene i dati relativi a chi ha scritto il libro

Attributi

- Cognome_autore: *stringa* <primary_key>
- Nome_autore: *stringa*
- Luogo_nascita_autore: *stringa*
- Data_nascita_autore: *data*

2.1.4 Utente

L'entità utente contiene le informazioni relative a chi effettuerà il prestito di un libro

Attributi

- Matricola: *intero* - <primary key>
- Nome_utente: *stringa*
- Cognome_utente: *stringa*
- Iniziali_sn_utente: *carattere*
- Data_n_utente: *data*

2.1.5 Editore

L'entità editore contiene le informazioni relative alla casa editrice che ha pubblicato il libro

Attributi

- Codice_editore: *intero* - <primary key>
- Nome_editore: *stringa*
- Telefono_editore: *intero*
- Via_editore: *stringa*
- Città_editore: *stringa*
- CAP_editore: *intero*

2.1.6 Copie libro

Questa entità si è creata onde evitare ridondanze per quanto riguarda la differenziazione tra l'ISBN e l'ID del libro. Contiene quindi tutte le informazioni per quanto riguarda ogni singola copia di ogni libro

Attributi

- ID: *intero* - <primary key>
- ISBN - *stringa*
- Cod_biblioteca - *intero*
- Presenza - *boolean*

2.1.7 Scritto da

Entità tramite tra libro e autore, permette di evitare ridondanze all'interno della base dati

Attributi

- Isbn: *stringa*
- Cognome_autore: *stringa*

2.1.8 Prestito

Qui troviamo tutte le informazioni relative ad un prestito

Attributi

- ID_libro: *intero*
- Matricola: *intero*
- Cod_biblioteca: *intero*
- Data_prestito: *data*
- Scadenza_prestito: *data*
- Data_consegna: *data*
- Consegnato: *booleano*

2.2 Descrizione testuale delle relazioni tra entità

2.2.1 Biblioteca – Libro: “Copie_libro”

Molteplicità N:M In una biblioteca possono trovarsi 1 o più libri (con molteplicità 1:N), mentre un libro può essere trovato in più biblioteche, in molteplici copie, o non trovarvisi affatto (con molteplicità 0:N). Ne consegue che la molteplicità della relazione sia N:M. Per via della sua molteplicità, abbiamo relazionato le due entità creando una nuova entità COPIE_LIBRO (descritta già sopra) e inserendo come attributi l'ID identificativo per ogni singola copia di ogni libro, l'ISBN al quale ogni copia si riferisce e il codice della biblioteca nel quale ogni singola copia si trova. In aggiunta per comodità è stato aggiunto il campo Presenza, attributo di tipo Booleano per tenere traccia della presenza di un libro in una certa biblioteca, che sarà sempre di valore 1 fintanto che vi si trova, per poi assumere valore 0 qualora venisse preso in prestito. Pertanto quest'ultimo campo verrà aggiornato qualora avvenga un cambiamento nella sezione Prestito che vedremo in seguito.

2.2.1 Autore – Libro: “Scritto_da”

Molteplicità N:M Un libro può essere scritto da uno o più autori (con molteplicità pari a 1:N), mentre un autore può scrivere uno o più libri (con molteplicità 1:N). Ne consegue che la molteplicità della relazione sia N:M, creandosi così una nuova entità SCRITTO_DA, contenente per l'appunto i campi ISBN e Cognome_autore.

2.2.1 Utente – Copie_libro: “Prestito”

Molteplicità N:M Un libro può essere preso in prestito una o più volte (con molteplicità 1:N), mentre un utente può prendere in prestito 1 o più libri (con molteplicità 1:N). Ne consegue che la molteplicità della relazione sia N:M. Si crea così l'entità Prestito contenente non solo questa volta le informazioni di associazione delle suddette entità, ma avremo la presenza di campi essenziali per un prestito, quali la sua data d'inizio, quella della scadenza, la data di consegna e un dato di tipo booleano (Consegnato) per tenere traccia dello stato di avanzamento del prestito. Aprendo una parentesi su quest'ultimo attributo, esso assumerà valore 0 fintanto che un libro rimarrà in prestito ad un utente, mentre assumerà valore 1 qualora un utente lo riconsegnasse.

2.2.1 Editore – Libro

Molteplicità 1:N Un libro può essere pubblicato da una e una sola casa editrice (con molteplicità 1:1), mentre un editore può pubblicare 1 o più libri (con molteplicità 1:N). Ne consegue che la molteplicità della relazione sia 1:N. Associamo quindi il codice dell'editore all'entità LIBRO, avendo il campo aggiuntivo <codice_editore> che si riferirà direttamente alla casa editrice che ha pubblicato il libro.

2.3 Modello E/R

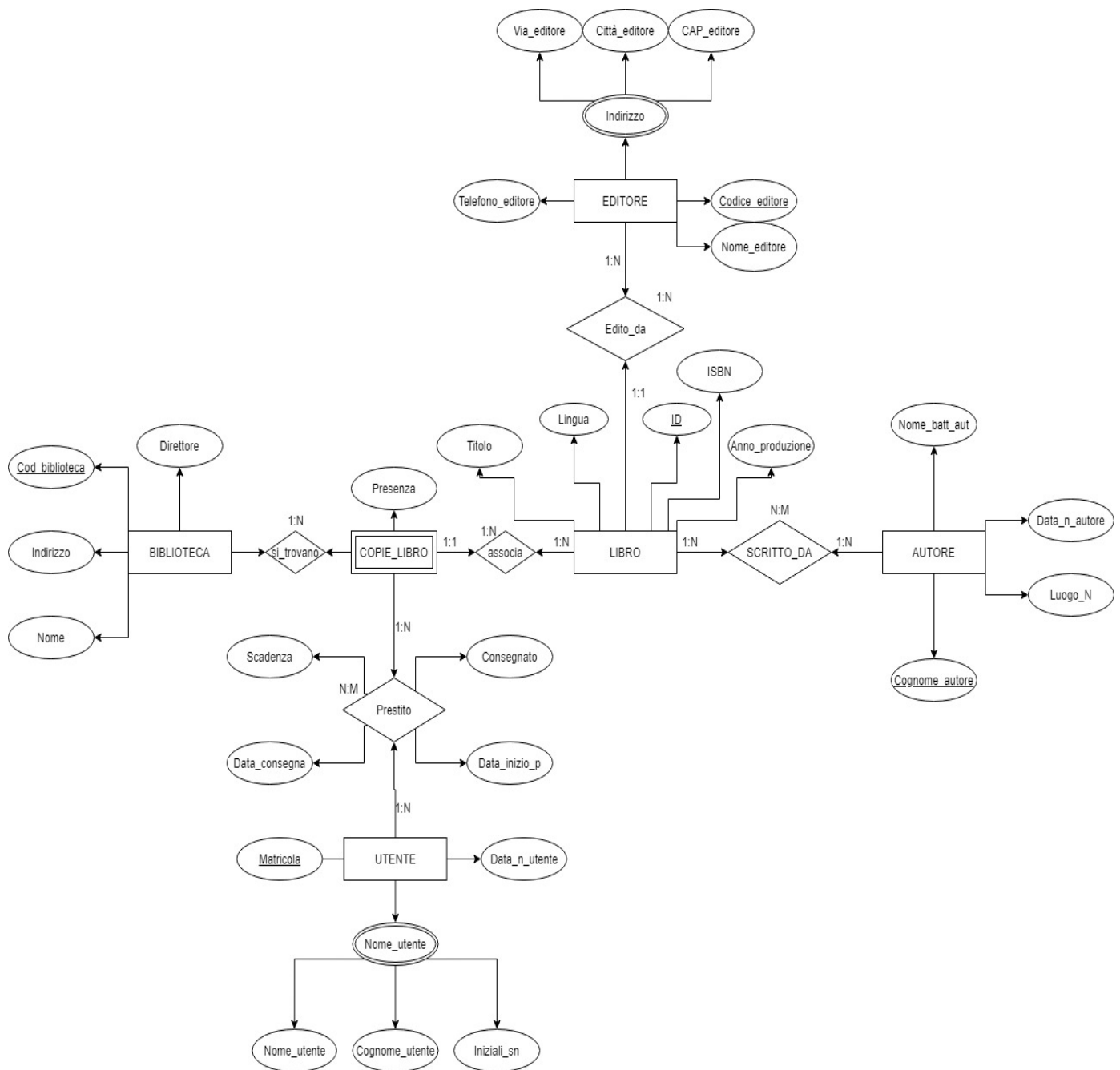


Figura 1: Modello E/R rappresentativo della base di dati

Possiamo notare in figura che esistono altre due relazioni collegate a **COPIE_LIBRO** non menzionate nelle descrizioni testuali. Esse (considerando **COPIE_LIBRO** come entità debole) si vengono a creare per associare la suddetta entità a **BIBLIOTECA** ed a **LIBRO**, entrambe con molteplicità 1:N. Ne consegue che entrambe si collegheranno per l'appunto a **COPIE_LIBRO** usufruendo della chiave primaria come appoggio per le relazioni

2.4 Schema Relazionale

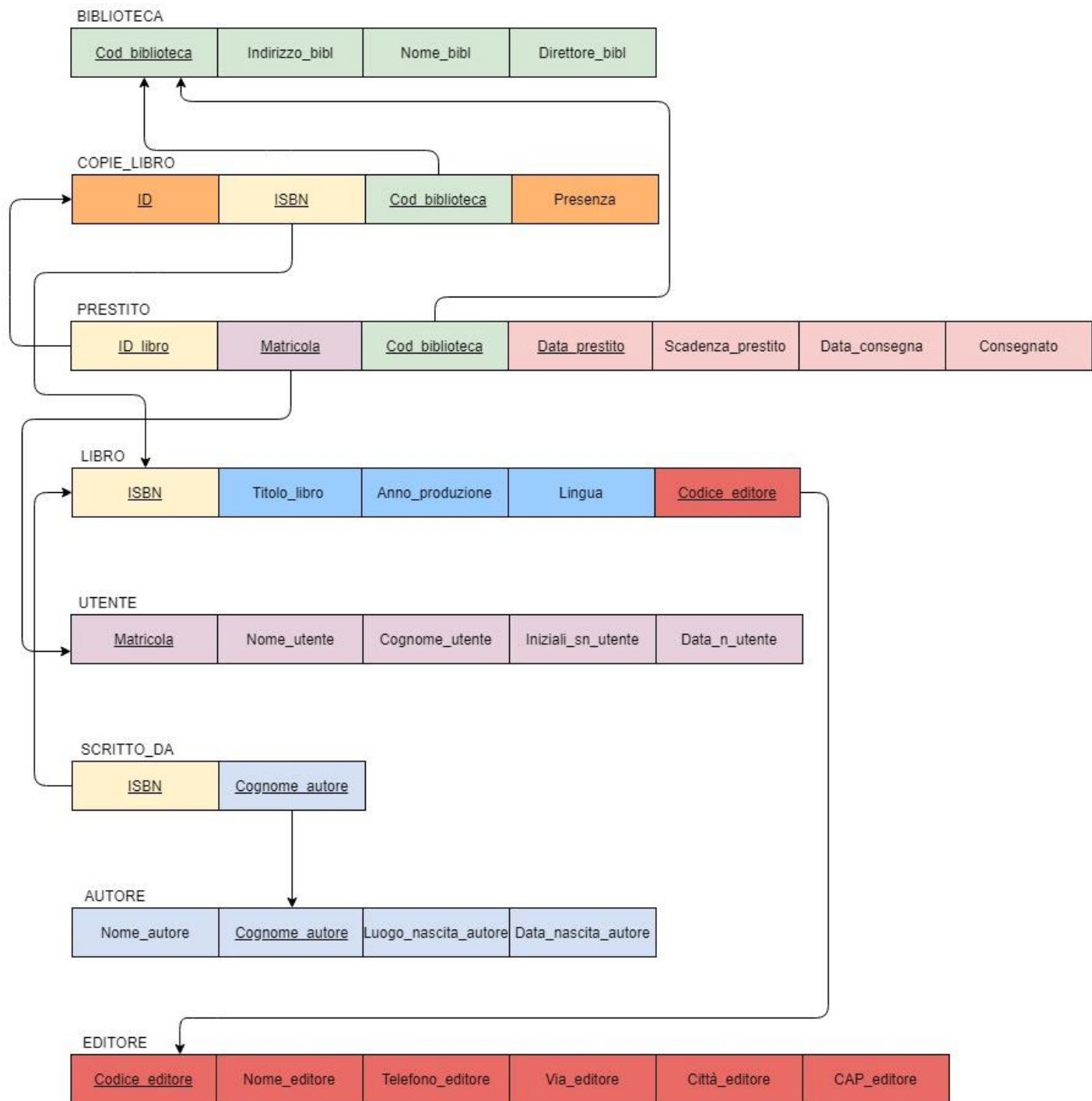


Figura 2: Schema relazionale rappresentativo della base di dati

Per una piccola precisazione ci teniamo ad avvisare che le chiavi esterne contenute all'interno di questo schema sono state lasciate volutamente sottolineate (ad esempio ISBN in COPIE_LIBRO) nonostante non siano chiave primaria di quell'entità, ma di quella a cui si riferisce

2.5 Normalizzazione Schema Relazionale

Una volta creato il diagramma E/R e convertitolo in uno schema relazionale, si ha avuto necessità di eliminare tutte le “anomalie” da esso, ad esempio ridondanze che andavano ad inserire inutilmente informazioni già presenti all’interno della base di dati. Per farlo, e per ottenere uno schema visivamente e concettualmente più pulito, bisognava ricorrere alla normalizzazione e di conseguenza applicare allo schema già presente le 3 forme normali più comunemente usate allo scopo.

2.5.1 Prima Forma Normale (1NF)

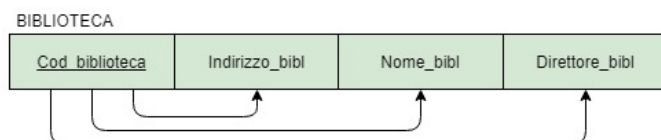
Una volta creato lo schema, sono stati rimossi tutti gli attributi multivalore in modo da rendere gli attributi rimasti “Atomici”. I candidati per farlo erano quindi l’indirizzo, per quanto riguarda l’editore, e il nome utente per quanto riguarda invece l’utente (come possiamo vedere in figura 1). Nello Schema Relazionale finale infatti questi attributi multivalore sono stati convertiti in 3 attributi a se stanti ciascuno:



2.5.2 Seconda e Terza Forma Normale (2NF/3NF)

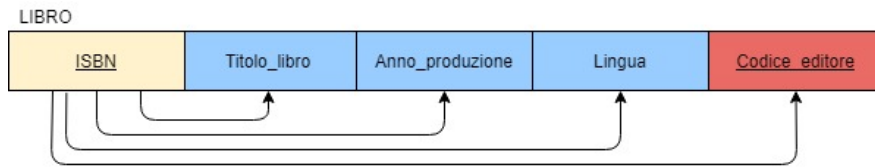
Il secondo passo per la normalizzazione era quello di assicurarsi che ogni attributo non primo fosse totalmente dipendente dalla chiave primaria (2NF) e che non fosse dipendente da una qualsiasi chiave non primaria (3NF) di una relazione. Precisiamo che non saranno presenti le relazioni COPIE_LIBRO, SCRITTO_DA e PRESTITO in quanto contenenti solamente chiavi esterne che relazionano due entità con al più qualche attributo essenziale per le informazioni contenute in esse. Procediamo quindi a verificare che effettivamente ogni entità rispetti quanto detto, analizzando una per una ognuna di esse:

- BIBLIOTECA



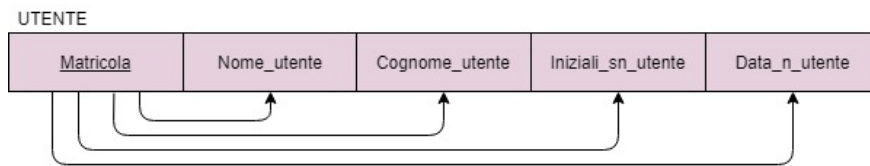
Qui possiamo notare che l’indirizzo, il nome e il direttore della biblioteca sono tutti e tre dipendenti dal codice. Ne consegue quindi che i 3 attributi non chiave sono totalmente dipendenti dalla chiave primaria, dichiarando quindi rispettata la seconda forma normale. Inoltre possiamo notare non essere dipendenti tra di loro, dichiarando rispettata anche la terza forma normale.

- LIBRO



Anche in questo caso possiamo notare che il titolo, l'anno la lingua e l'editore sono tutti totalmente dipendenti dall'ISBN del libro e da nessun altro attributo chiave o non chiave, confermando sia le regole della seconda che della terza forma normale.

- UTENTE



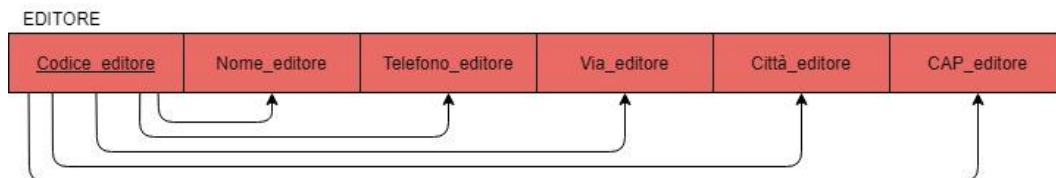
Utente ha attributi semplici legati all'identificatore matricola, ne consegue che ogni attributo è dipendente da matricola, rispettando così la seconda forma normale, nonché non dipendente da altri attributi non chiave, rispettando così anche la terza forma normale.

- AUTORE



Autore così come Utente ha informazioni basiche legate al cognome, ne consegue che ognuna di queste informazioni sono completamente dipendenti dal cognome e da nessun altro attributo non chiave, rispettando così sia la seconda che la terza forma normale.

- EDITORE



Editore è identificato unicamente dal codice, che rappresenta l'unico attributo chiave da cui dipendono tutti gli altri attributi di questa relazione. Anche in questo caso sia la seconda che la terza forma normale sono rispettate.

Capitolo 3

Costruzione

3.1 Introduzione

Partendo dalle basi, abbiamo scelto MySQL Workbench come tool per la creazione del database e dei file SQL necessari per la base di dati, procedendo a creare le tabelle citate nei capitoli precedenti. Una volta create le tabelle con le apposite query, abbiamo manipolato il file .CSV fornitoci tramite file script utilizzando la shell di Linux, convertendo poi il risultato in query SQL tramite un tool online. Successivamente abbiamo popolato nostra base dati, e definito le query per le interrogazioni alla base di dati (elencate in seguito), per verificarne il corretto funzionamento.

3.2 Interrogazioni della base di dati

Di seguito approfondiremo e analizzeremo una per una ognuna delle interrogazioni della nostra base di dati, sia quelle richieste nei requisiti minimi che quelle aggiuntive da noi introdotte

3.2.1 Statistiche di base in algebra relazionale

L'elenco di tutti i libri dati in prestito ad un utente

$$\text{MATRICOLE_UTENTI} \leftarrow \Pi_{\text{Matricola}, \text{ID}} (\text{UTENTE} \bowtie_{\text{UTENTE.Matricola} = \text{Prestito.Matricola AND UTENTE.Matricola} = \text{'Input' PRESTITO}})$$
$$\text{LIBRI_UTENTI} \leftarrow \Pi_{\text{Matricola}, \text{ID}, \text{ISBN}} (\text{MATRICOLE_UTENTI} \bowtie_{\text{MATRICOLE_UTENTI.ID} = \text{COPIE_LIBRO.ID COPIE_LIBRO}})$$
$$\text{RESULT} \leftarrow \Pi_{\text{Titolo}, \text{ID}, \text{Matricola}} (\text{LIBRI_UTENTI} \bowtie_{\text{LIBRI_UTENTI.ISBN} = \text{LIBRO.ISBN LIBRO}})$$

L'elenco di tutti gli utenti che hanno preso in prestito un determinato libro

$$ID_BOOK \leftarrow \Pi_{ID, Titolo_libro} (LIBRO \bowtie_{LIBRO.ISBN = COPIE_LIBRO.ISBN \text{ AND } LIBRO.Titolo_libro = 'Input' COPIE_LIBRO})$$

$$MATR_UTENTI \leftarrow \Pi_{ID, Matricola, Titolo_libro} (ID_BOOK \bowtie_{ISBN_BOOK.ID = PRESTITO.ID} PRESTITO)$$

$$RESULT \leftarrow \Pi_{ID, Titolo_libro, Matricola, Nome, Cognome} (MATR_UTENTI \bowtie_{MATR_UTENTI.Matricola = UTENTE.Matricola} UTENTE)$$

Determinare l'autore che ha scritto più libri

$$AUTOR \leftarrow \Pi_{Cognome_autore, ISBN} (AUTORE \bowtie_{AUTORE.Cognome_autore = SCRITTO_DA.Cognome_autore} SCRITTO_DA)$$

$$AUT1(A1, C1) \leftarrow \text{Cognome_autore } \mathbf{f}COUNT(ISBN) (AUTOR)$$

$$AUT2(A2, C2) \leftarrow AUT1$$

$$NO_GOOD \leftarrow AUT1 \bowtie_{C1 < C2} AUT2$$

$$RES \leftarrow AUT1 - NO_GOOD$$

Determinare l'editore che ha pubblicato più libri

$$EDITOR \leftarrow \Pi_{Nome_editore, ISBN} (EDITORE \bowtie_{EDITORE.Codice_editore = LIBRO.Codice_editore} LIBRO)$$

$$EDITOR1(E1, C1) \leftarrow \text{Nome_editore } \mathbf{f}COUNT(ISBN) (EDITOR)$$

$$EDITOR2(E2, C2) \leftarrow EDITOR1$$

$$NO_GOOD \leftarrow EDITOR1 \bowtie_{C1 < C2} EDITOR2$$

$$RES \leftarrow EDITOR1 - NO_GOOD$$

Determinare le 5 lingue più comuni in cui sono scritti i libri presenti nella basi di dati.

Per questa query particolare ci siamo limitati a trovare la lingua più comune utilizzata, in quanto facendo alcune ricerche, con nostro dispiacere non siamo riusciti a trovare una soluzione che emulasse il limit 5 delle query in sql.

LINGUE1(L1,C1)←Lingua \mathbf{fCOUNT} (Lingua) (LIBRO)

LINGUE2(L2,C2)← LINGUE1(L1,C1)

NO_GOOD←LINGUE1 \bowtie C1<C2 LINGUE2

RES←LINGUE1 – NO_GOOD

3.2.2 Interrogazioni aggiuntive

1) Accettato in input un autore, visualizzare la casa editrice più utilizzata da esso per la pubblicazione dei libri

ISBN_COG← Π_{ISBN} (AUTORE \bowtie AUTORE.Cognome_autore = SCRITTO_DA.Cognome_autore AND
AUTORE.Cognome_autore = 'Input' SCRITTO_DA)

CODICI← $\Pi_{Codice_editore}$ (ISBN_COG \bowtie ISBN_COG.ISBN = LIBRO.ISBN
LIBRO)

EDIT← $\Pi_{Nome_editore}$ (CODICI \bowtie CODICI.Codice_editore = EDITORE.Codice_editore EDITORE)

EDITOR1(E1,C1)←Nome_editore \mathbf{fCOUNT} (Nome_editore) (EDIT)

EDITOR2(E2,C2) ← EDITOR1(E1,C1)

NO_GOOD←EDITOR1 \bowtie C1<C2 EDITOR2

RES←EDITOR1 – NO_GOOD

2) Visualizzare tutti gli utenti che hanno consegnato un libro in ritardo, incluso il rispettivo libro

UT_RITARDO← $\Pi_{Nome_utente, Cognome_utente, Matricola}$ (PRESTITO \bowtie
PRESTITO.Matricola = UTENTE.Matricola AND PRESTITO.Consegnato = 1
UTENTE)

UT_RITARDO2← $\Pi_{Nome_utente, Cognome_utente, Matricola, ISBN}$ (UT_RITARDO \bowtie
UT_RITARDO.ID = COPIE_LIBRO.ID COPIE_LIBRO)

RES← $\Pi_{Nome_utente, Cognome_utente, Matricola, Titolo_libro}$ (UT_RITARDO2 \bowtie
UT_RITARDO2.ISBN = LIBRO.ISBN LIBRO)

3) Visualizzare tutti gli utenti che hanno preso in prestito libri di autori Newyorkesi

$$\text{UTENTI1} \leftarrow \Pi_{\text{ID, Nome_utente, Cognome_utente, Matricola}} (\text{PRESTITO} \bowtie \\ \text{PRESTITO.Matricola} = \text{UTENTE.Matricola AND PRESTITO.Consegnato} = 1 \\ \text{UTENTE})$$

$$\text{UTENTI2} \leftarrow \Pi_{\text{ID, Nome_utente, Cognome_utente, Matricola, ISBN}} (\text{UTENTI1} \bowtie \\ \text{UTENTI1.ID} = \text{COPIE_LIBRO.ID UTENTE})$$

$$\text{UTENTI3} \leftarrow \Pi_{\text{ID, Nome_utente, Cognome_utente, Matricola, Cognome_autore}} (\text{UTENTI2} \bowtie \\ \text{UTENTI2.ISBN} = \text{SCRITTO_DA.ISBN SCRITTO_DA})$$

$$\text{RES} \leftarrow \Pi_{\text{ID, Nome_utente, Cognome_utente, Matricola}} (\text{UTENTI3} \bowtie \\ \text{UTENTI3.Cognome_autore} = \text{AUTORE.Cognome_autore AUTORE})$$

4) Vista la necessità, una delle interrogazioni aggiuntive create è quella della ricerca avanzata, tenendo conto che vengano riempiti tutti i campi del form. Di seguito la query in algebra relazionale:

$$\text{BOOK1} \leftarrow \Pi_{\text{ID, ISBN, Lingua, Anno_pubblicazione, Presenza, Cod_biblioteca}} (\text{COPIE_LIBRO} \bowtie \\ \text{COPIE_LIBRO.ISBN} = \text{LIBRO.ISBN AND LIBRO.Anno_pubblicazione} > \\ \text{'InputAnno1'} \text{ AND LIBRO.Anno_pubblicazione} < \text{'InputAnno2'} \text{ LIBRO})$$

$$\text{BOOK2} \leftarrow \Pi_{\text{ID, ISBN, Lingua, Anno_pubblicazione, Presenza, Indirizzo_biblioteca, Nome_biblioteca}} \\ (\text{BOOK1} \bowtie \text{BOOK1.Cod_biblioteca} = \text{BIBLIOTECA.Cod_biblioteca} \\ \text{AND BIBLIOTECA.Nome_biblioteca} = \text{'InputBiblioteca'} \text{ BIBLIOTECA})$$

$$\text{BOOK3} \leftarrow \Pi_{\text{ID, ISBN, Lingua, Anno_pubblicazione, Presenza, Indirizzo_biblioteca, Nome_biblioteca, Cognome_autore}} \\ (\text{BOOK2} \bowtie \text{BOOK2.ISBN} = \text{SCRITTO_DA.ISBN} \\ \text{AND SCRITTO_DA.Cognome_autore} = \text{'InputCogAutore'} \text{ SCRITTO_DA})$$

$$\text{BOOK4} \leftarrow \Pi_{\text{ID, ISBN, Lingua, Anno_pubblicazione, Presenza, Indirizzo_biblioteca, Nome_biblioteca, \\ Nome_autore, Cognome_autore}} \\ (\text{BOOK3} \bowtie \text{BOOK3.Cognome_autore} = \text{AUTORE.Cognome_autore} \\ \text{AND AUTORE.Nome_autore} = \text{'InputNomeAutore'} \text{ AUTORE})$$

$$\text{RES} \leftarrow \Pi_{\text{ID, ISBN, Lingua, Anno_pubblicazione, Presenza, Indirizzo_biblioteca, Nome_biblioteca, \\ Nome_autore, Cognome_autore, Nome_editore}} \\ (\text{BOOK4} \bowtie \text{BOOK4.Codice_editore} = \text{EDITORE.Codice_editore} \\ \text{AND EDITORE.Nome_editore} = \text{'InputEditore'} \text{ EDITORE})$$

All'interno del sito sono presenti altre interrogazioni di minore entità che non vengono menzionate qui, ma verranno citate nel capitolo della costruzione del sito (parte php).

Capitolo 4

Realizzazione

In questa sezione andremo a trattare gli aspetti e le scelte che riguardano la realizzazione del sito web

4.1 HTML/CSS/PHP

4.1.1 HTML

Il linguaggio di marcatura adottato è HTML 5. Questa tecnologia permette di creare documenti strutturati tramite l'utilizzo di una semantica che non richiede l'utilizzo di fogli di stile, anche se questi sono stati implementati per usufruire in maniera più moderna delle varie voci del sito. I file creati riportano:

```
<!DOCTYPE HTML>
<html>
```

Dove DOCTYPE indica la definizione del linguaggio di marcatura e permette di uniformare il rendering fra i vari browser; questo tuttavia non è sempre vero come nel caso si usi Internet Explorer 6.

Form

I form presenti sono caratterizzati da un tipo di input variabile: infatti i tipi di dato spaziano da testuale, numerico e data. Questa scelta è compatibile con i browser più utilizzati per Windows e Linux, ma che presenta dei problemi per quanto riguarda Safari; questo infatti non implementa ad oggi il supporto al tag di input di tipo **date**. Questa scelta è stata ritenuta accettabile poiché in un contesto universitario è meno probabile che le biblioteche operino con terminali sviluppati da Apple e quindi con Safari. I form con elementi di input obbligatori sono marcati con **required** che assicura che il campo su cui si presenta sia compilato. Nei file di modifica è stato scelto di non permettere la modifica dei valori corrispondenti alle chiavi primarie e per fare questo è stato usato l'attributo **disable**

4.1.2 CSS

Il sito sfrutta il framework open-source **Bootstrap**, che contiene una raccolta di modelli di progettazione basati su HTML e CSS; Bootstrap viene implementato tramite link a server remoti. Questa scelta:

- alleggerisce il carico di richieste che si devono gestire, poiché chi visita il sito ha già Bootstrap nella propria cache non dovrà scaricarlo nuovamente
- permette la creazione di pagine **responsive**, cioè il cui layout si ridimensiona in base al dispositivo o monitor utilizzato

Il sito utilizza anche un foglio di stile locale, style.css, che si occupa di definire degli stili per migliorare l'esperienza di utilizzo dell'applicazione, come popup di alert, oppure di ridefinire stili già esistenti utilizzati da bootstrap.



4.1.3 PHP

Database

La connessione al database Gestione_biblioteca si effettua tramite **mysqli_connect** a cui vengono passati indirizzo dell'host, nome dell'user, eventuale password e nome del database.

Dinamicità

Il php ha un ruolo di primaria importanza nel progetto, infatti questo linguaggio lato server ha permesso la creazione di un sito dinamico; è per esempio possibile inserire, modificare, archiviare, visualizzare un prestito attraverso l'apposita sezione "Operazioni". La barra di navigazione e il footer del sito sono stati posti in file separati, poi richiamati negli altri file; inoltre un file "funzioni" contiene codici che appaiono in più parti del sito, sempre per evitare di duplicare il codice nelle varie pagine.

Il gruppo ha dato inoltre importanza al controllo dei dati inseriti. Per quanto riguarda i prestiti, durante l'inserimento di un prestito, si viene subito avvisati se la matricola o l'ID del libro inseriti esistono; altri controlli sono:

- limite di 5 libri presi in prestito al giorno
- disponibilità o meno del libro
- lo stesso libro non può essere preso 2 volte lo stesso giorno dallo stesso utente

Durante la registrazione di un utente invece, è necessario che l'utente sia maggiorenne (non vi sono universitari minorenni).

Una funzione essenziale in ambito di gestione prestiti è quella di mostrare lo stato di un prestito di un utente. È presente infatti controllare in ogni momento quanto tempo manca alla consegna di un libro, o di quanti giorni è in ritardo un utente nella consegna dello stesso

Cerca libro

Ricerca avanzata

Cerca

Studenti in ritardo

Antonio Russo (444444) in ritardo di 108 giorni.
Vincenzo Fontana (888888) in ritardo di 49 giorni.
Rosa Bianchi (555555) in ritardo di 64 giorni.
Francesca Rizzo (999999) in ritardo di 20 giorni.
Rosa Bianchi (555555) in ritardo di 10 giorni.

Mostra altro

Statistiche

Utenti che hanno consegnato un prestito in ritardo

Titolo libro	Cognome utente	Nome utente	Matricola
Arthur and	Bianchi	Rosa	555555

FIGURA 1 SCHERMATA PRINCIPALE

4.2 Accessibilità

4.2.1 Barra di ricerca

Abbiamo implementato nei file che visualizzano una mole elevata di dati (o comunque possibilmente elevata), una barra per cercare e filtrare tramite matricola i risultati delle varie interrogazioni

Filtra per matricola

Cerca

La barra di ricerca principale presente nella home permette di ricercare per titolo del libro e per autore, è inoltre presente una ricerca avanzata in cui è possibile affinare la ricerca:

Cerca libro

Ricerca avanzata

Intervallo date

editore

biblioteca

Cerca

Il risultato di questo form permetterà di sapere oltre al luogo in cui si trova il libro, anche se è disponibile

4.2.2 Struttura e comportamento

Amministratore

Il sito è accessibile solamente dagli utenti admin, che possono svolgere qualsiasi operazione di modifica ed eliminazione dei dati. Di seguito una breve descrizione delle sezioni in cui può operare un amministratore:

- **Navbar e Footer:** Tutte le schede prevedono una barra di navigazione con dropdown, che permette di raggiungere tutte le funzionalità; il footer contiene informazioni utili riguardo l'ambito accademico nella sua generalità.
- **Home:** Schermata con la barra di ricerca principale, una visualizzazione sintetica dello stato dei prestiti in ritardo ed alcune statistiche che riguardano il database.
- **Utenti:** dropdown con 2 voci:
 - **Registrazione:** utilizzata per inserire un nuovo utente
 - **Elenco utenti:** utilizzata per visualizzare gli utenti presenti nel sistema
- **Prestiti:** dropdown con 3 voci:
 - **Inserisci prestito:** utilizzata per inserire un nuovo prestito
 - **Prestiti attivi:** utilizzata per visualizzare lo stato dei prestiti
 - **Archivio prestiti:** mostra tutti i prestiti completati(??va bene ??)
- **Statistiche:** dropdown con 4 voci:
 - **Libri dati in prestito ad un utente:** dato il numero di matricola, mostra tutti i libri da esso presi in prestito
 - **Utenti con un determinato libro:** dato il titolo di un libro, mostra tutti gli utenti che hanno attualmente in prestito alcune copie di quel libro
 - **Case editrici più usate:** dato un autore, mostra la casa editrice da lui utilizzata più spesso per la pubblicazione dei suoi libri
 - **Statistiche base:** nelle statistiche base abbiamo statistiche di interrogazioni comuni del database, quali le 5 lingue più comuni, l'autore/editore che ha scritto/pubblicato più libri, l'elenco degli utenti che hanno consegnato un libro in ritardo e l'elenco degli utenti che hanno preso in prestito libri di autori nati a New York