Biblioteca

Capucci Fabio

Matricola: 757969

fabio.capucci@studio.unibo.it

Cortesi Emanuele

Matricola: 758226

emanuele.cortesi2@studio.unibo.it

Fabbri Matteo

Matricola: 753186

matteo.fabbri27@studio.unibo.it

Sommario

[1) Analisi dei requisiti 2](#_Toc503886541)

[1a) Requisiti espressi in linguaggio naturale 2](#_Toc503886542)

[1b) Glossario dei termini 2](#_Toc503886543)

[1c) Specifica operazioni 3](#_Toc503886544)

[2) Progettazione concettuale 3](#_Toc503886545)

[2a) Identificazione delle entità e relazioni 3](#_Toc503886546)

[2b) Schema Entity-Relationship 4](#_Toc503886547)

[2c) Regole 4](#_Toc503886548)

[3) Progettazione logica 5](#_Toc503886549)

[3a) Ristrutturazione dello schema concettuale 5](#_Toc503886550)

[3b) Normalizzazione 5](#_Toc503886551)

[3c) Schema Relazionale 5](#_Toc503886552)

[4) SQL 6](#_Toc503886553)

[4a) Query CREATE 6](#_Toc503886554)

[4b) Query INSERT 8](#_Toc503886555)

[4c) Query UPDATE 9](#_Toc503886556)

[4d) Query SELECT 9](#_Toc503886557)

Il sito web per visualizzare il progetto è: <http://site1803.tw.cs.unibo.it/>

# 1) Analisi dei requisiti

## 1a) Requisiti espressi in linguaggio naturale

Si vuole realizzare un database per la biblioteca cittadina in grado di semplificare l'organizzazione delle varie attività. Per quanto riguarda i libri verranno memorizzati tramite un id univoco, il codice ISBN, la quantità di copie dello stesso volume ed il suo titolo. Per gli autori saranno necessari un nome, un cognome ed un identificativo (id). Ci saranno poi le collane dei vari libri che verranno registrate con una descrizione un id ed il loro nome. La posizione dei libri sarà caratterizzata dalla stanza e da un espositore in cui si trovano. La casa editrice avrà il suo nome ed un codice (id). Per i generi letterali si vorrà sapere nome ed identificativo. Il personale che ha l'unico compito della gestione dei prestiti dei libri sarà descritto da il proprio nome e cognome, la data di nascita ed un id. Gli utenti saranno registrati con il loro nome, cognome, città ed indirizzo di residenza, data di nascita ed un codice (id). Occorrerà registrare anche la totalità dei prestiti di libri effettuati verso gli utenti mediante l'intervento del personale, descrivendo quest'operazione tramite la data di inizio, la data di scadenza (o data di fine), un campo "restituito" per indicare se la restituzione del prestito è andata a buon fine ed un id univoco.

## 1b) Glossario dei termini

|  |  |  |
| --- | --- | --- |
| Termine | Descrizione | Collegamenti |
| Libri | Elenco di tutti i libri che fanno parte dell'inventario della biblioteca | Autori, Collane, Posizioni libri, Casa editrice, Generi, Personale, Utenti, Prendono in prestito |
| Autori | Scrittori dei vari libri, un libro può essere scritto anche da più autori | Libri |
| Collane | Sono una serie di libri riguardanti il medesimo contesto narrativo | Libri |
| Posizione libri | Descrivono la posizione esatta di ogni libro, se un libro è presente in molteplici copie sono sempre considerate raggruppate nella stessa pozione, specifica la stanza e l’espositore nel quale si trova il libro | Libri |
| Casa editrice | Enti che si occupano della pubblicazione dei libri | Libri |
| Generi | I vari generi narrativi nei quali sono caratterizzati i libri, un libro può appartenere anche a più generi | Libri |
| Personale | L'insieme dei dipendenti della biblioteca che hanno lo scopo di essere intermediari nei prestiti tra i libri e gli utenti | Libri, Prendono in prestito, Utenti |
| Utenti | L'insieme delle persone fisiche che si recano in biblioteca per chiedere i libri in prestito | Libri, Personale, Prendono in prestito |
| Prendono in prestito | L'attività del concedere un libro in prestito | Libri, Personale, Utenti |

## 1c) Specifica operazioni

Per questa fase verrà presa in esame una biblioteca avente sede in una città di medie dimensioni, come Ravenna

1. Inserire un nuovo autore (in media 25 volte alla settimana)
2. Inserire un nuovo libro (in media 50 volte alla settimana)
3. Inserire una nuova collana (in media 5 volte alla settimana)
4. Inserire una nuova posizione libro (in media 5 volte l’anno)
5. Inserire una nuova casa editrice (in media 5 volte l'anno)
6. Inserire un nuovo genere (in media 3 volte l'anno)
7. Inserire un nuovo membro del personale (in media 2 volte l'anno)
8. Inserire un nuovo utente (in media 5 volte al giorno)
9. Modificare la posizione dei libri (in media 50 volte al mese)
10. Registrare un nuovo prestito (in media 50 volte al giorno)
11. Visualizzare l'elenco dei libri presi in presti da un utente (in media 50 volte al giorno)
12. Visualizzare i libri scritti da un certo autore (in media 25 volte al giorno)
13. Visualizzare i libri che appartengo ad un certo genere (in media 20 volte a giorno)
14. Visualizzare i libri presi in prestito non ancora restituiti oltre la scadenza (in media 5 volte al giorno)
15. Visualizzare il numero dei libri presi in prestito in un determinato periodo di tempo (in media 2 volte al giorno)
16. Visualizzare la posizione di un determinato libro (in media 150 volte al giorno)

# 2) Progettazione concettuale

## 2a) Identificazione delle entità e relazioni

Per quanto riguarda la progettazione concettuale siamo partiti analizzando il fulcro centrale del nostro database: i libri, gli utenti e i prestiti.

Prestito

Utenti

Libri

Successivamente abbiamo aggiunto altre informazioni dei libri come gli autori, la casa editrice, il genere e la collana.

Autori

Scrivono

Appartengono

Hanno

Libri

Pubblica

Generi

Collana

Casa editrice

In fine abbiamo aggiunto una posizione del libro: divisa in stanza ed espositore di appartenenza ed il personale addetto ai prestiti dei vari libri.

Abbiamo così ottenuto il seguente schema Entity-Relationship.

## 2b) Schema Entity-Relationship

## 2c) Regole

|  |
| --- |
| **Regole di vincolo** |
| I libri che sono disponibili in molteplici copie sono sempre considerati nella medesima posizione |
| L’attributo quantità di libri fa rifermento ai libri totali della biblioteca e non a quelli disponibili |
| La quantità di un libro uguale a zero equivale a dire che tale libro non è più presente all’interno della biblioteca |
| Un utente può avere al massimo tre prestiti attivi contemporaneamente |
| Per la creazione di un prestito un utente si reca da un membro del personale con il libro sul quale creare il prestito; l’addetto legge l’id del libro dall’etichetta sul libro, l’id dell’utente dalla tessera dell’utente e il suo id. In tal modo non è possibile creare un prestito di un libro non presente in bibliteca |

# 3) Progettazione logica

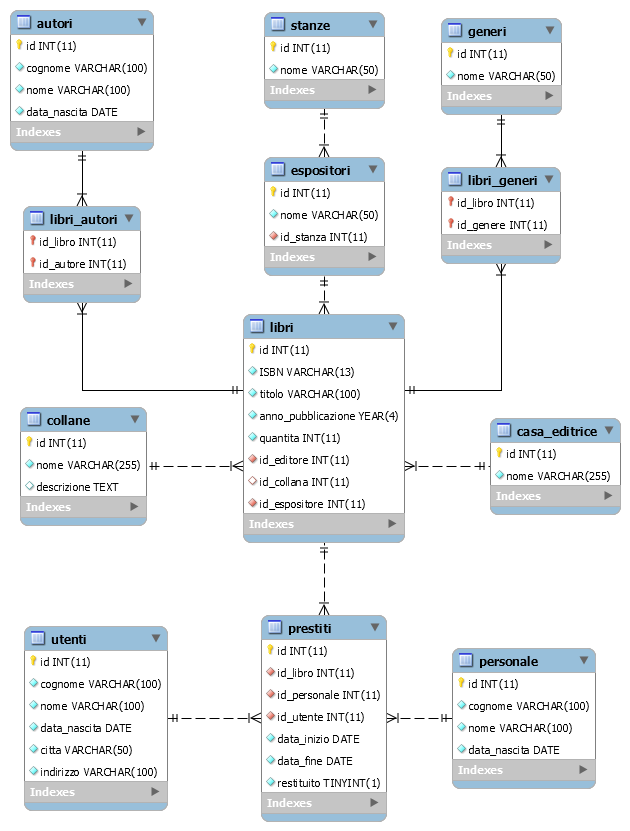
## 3a) Ristrutturazione dello schema concettuale

Non sono state rilevate ridondanze perciò lo schema è rimasto il medesimo e lo stesso vale per la gestione degli accorpamenti e per gli attributi multivalore. L’identificatore primario è definito dall’attributo id.

## 3b) Normalizzazione

Il database è in terza forma normale perché è in seconda forma normale e tutti gli attributi non-chiave dipendono dalla chiave soltanto

## 3c) Schema Relazionale



# 4) SQL

## 4a) Query CREATE

**CREATE** TABLE collane (

Id INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(255) NOT NULL,

descrizione TEXT NULL,

PRIMARY KEY (id)

);

**CREATE** TABLE casa\_editrice (

Id INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(255) NOT NULL,

PRIMARY KEY (id)

);

**CREATE** TABLE autori (

Id INT NOT NULL AUTO\_INCREMENT,

cognome VARCHAR(100) NOT NULL,

nome VARCHAR(100) NOT NULL,

data\_nascita DATE NOT NULL,

PRIMARY KEY (id)

);

**CREATE** TABLE generi (

id INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(50) NOT NULL,

PRIMARY KEY (id)

);

**CREATE** TABLE stanze (

id INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(50) NOT NULL,

PRIMARY KEY (id)

);

**CREATE** TABLE espositori (

id INT NOT NULL AUTO\_INCREMENT,

nome VARCHAR(50) NOT NULL,

id\_stanza INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_stanza) REFERENCES stanze (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION

);

**CREATE** TABLE libri (

id INT NOT NULL AUTO\_INCREMENT,

ISBN VARCHAR(13) NOT NULL,

titolo VARCHAR(100) NOT NULL,

anno\_pubblicazione YEAR NOT NULL,

quantita INT NOT NULL,

id\_editore INT NOT NULL,

id\_collana INT NULL DEFAULT NULL,

id\_espositore INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (id\_editore) REFERENCES casa\_editrice (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION,

FOREIGN KEY (id\_collana) REFERENCES collane (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION,

FOREIGN KEY (id\_espositore) REFERENCES espositori (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION

);

**CREATE** TABLE libri\_autori (

id\_libro INT NOT NULL,

id\_autore INT NOT NULL,

PRIMARY KEY (id\_autore, id\_libro),

FOREIGN KEY (id\_autore) REFERENCES autori (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION,

FOREIGN KEY (id\_libro) REFERENCES libri (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION

);

**CREATE** TABLE libri\_generi (

id\_libro INT NOT NULL,

id\_genere INT NOT NULL,

PRIMARY KEY (id\_genere, id\_libro),

FOREIGN KEY (id\_libro) REFERENCES libri (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION,

FOREIGN KEY (id\_genere) REFERENCES generi (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION

);

**CREATE** TABLE personale (

id INT NOT NULL AUTO\_INCREMENT,

cognome VARCHAR(100) NOT NULL,

nome VARCHAR(100) NOT NULL,

data\_nascita DATE NOT NULL,

PRIMARY KEY (id)

);

**CREATE** TABLE utenti (

id INT NOT NULL AUTO\_INCREMENT,

cognome VARCHAR(100) NOT NULL,

nome VARCHAR(100) NOT NULL,

data\_nascita DATE NOT NULL,

citta VARCHAR(50) NOT NULL,

indirizzo VARCHAR(100) NOT NULL,

PRIMARY KEY (id)

);

**CREATE** TABLE prestiti (

id INT NOT NULL AUTO\_INCREMENT,

id\_libro INT NOT NULL,

id\_personale INT NOT NULL,

id\_utente INT NOT NULL,

data\_inizio DATE NOT NULL,

data\_fine DATE NOT NULL,

restituito BOOLEAN NOT NULL DEFAULT FALSE,

PRIMARY KEY (id),

FOREIGN KEY (id\_libro) REFERENCES libri (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION,

FOREIGN KEY (id\_personale) REFERENCES personale (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION,

FOREIGN KEY (id\_utente) REFERENCES utenti (id)

ON DELETE RESTRICT

ON UPDATE NO ACTION

);

## 4b) Query INSERT

**INSERT INTO** autori (id, cognome, nome, data\_nascita) VALUES (1, "Tolkien", "John Ronald Reuel", "1892-01-03");

**INSERT INTO** autori (id, cognome, nome, data\_nascita) VALUES (2, "Rowling", "Joanne", "1965-07-31");

**INSERT INTO** casa\_editrice (id, nome) VALUES (1, "Mondadori");

**INSERT INTO** casa\_editrice (id, nome) VALUES (2, "Zanichelli");

**INSERT INTO** collane (id, nome, descrizione) VALUES (1, "Il Signore degli Anelli", "Romanzo fantasy ambientato nella terra di mezzo. Parla di un Hobbit che deve distruggere un anello per portare la pace.");

**INSERT INTO** collane (id, nome, descrizione) VALUES (2, "Harry Potter", "Ogni libro della serie rappresenta un anno nella vita di Harry dagli undici ai diciassette anni; i libri descrivono ogni anno scolastico trascorso nella scuola di magia e stregoneria di Hogwarts.");

**INSERT INTO** generi (id, nome) VALUES (1, "Fantasy");

**INSERT INTO** generi (id, nome) VALUES (2, "Giallo");

**INSERT INTO** generi (id, nome) VALUES (3, "Avventura");

**INSERT INTO** personale (id, cognome, nome, data\_nascita) VALUES (1, "Rossi", "Mario", "1980-01-01");

**INSERT INTO** personale (id, cognome, nome, data\_nascita) VALUES (2, "Fabbri", "Matteo", "1996-05-04");

**INSERT INTO** utenti (id, cognome, nome, data\_nascita, citta, indirizzo) VALUES (1, "Cortesi", "Emanuele", "1996-09-30", "Bagnacavallo (RA)", "via aaa 1");

**INSERT INTO** utenti (id, cognome, nome, data\_nascita, citta, indirizzo) VALUES (2, "Capucci", "Fabio", "1996-01-16", "Fusignano (RA)", "via bbb 2");

**INSERT INTO** stanze(id, nome) VALUES (1, 'Per ragazzi');

**INSERT INTO** stanze(id, nome) VALUES (2, 'Per studenti');

**INSERT INTO** espositori(id, nome, id\_stanza) VALUES (1, 'A', '1');

**INSERT INTO** espositori(id, nome, id\_stanza) VALUES (2, 'B', '1');

**INSERT INTO** espositori(id, nome, id\_stanza) VALUES (3, 'A', '2');

**INSERT INTO** libri(id, ISBN, titolo, anno\_pubblicazione, quantita, id\_editore, id\_collana, id\_espositore)

VALUES (1, '1111111111111', 'La compagnia dellanello', 1980, 5, 1, 1, 1);

**INSERT INTO** libri\_autori(id\_libro, id\_autore) VALUES (1, 1);

**INSERT INTO** libri\_generi(id\_libro, id\_genere) VALUES (1, 1);

**INSERT INTO** libri\_generi(id\_libro, id\_genere) VALUES (1, 3);

**INSERT INTO** libri(id, ISBN, titolo, anno\_pubblicazione, quantita, id\_editore, id\_collana, id\_espositore)

VALUES (2, '2222222222222', 'Le due torri', 1985, 7, 1, 1, 1);

**INSERT INTO** libri\_autori(id\_libro, id\_autore) VALUES (2, 1);

**INSERT INTO** libri\_generi(id\_libro, id\_genere) VALUES (2, 1);

**INSERT INTO** libri\_generi(id\_libro, id\_genere) VALUES (2, 3);

**INSERT INTO** libri(id, ISBN, titolo, anno\_pubblicazione, quantita, id\_editore, id\_collana, id\_espositore)

VALUES (3, '3333333333333', 'La pietra filosofale', 1997, 9, 2, 2, 2);

**INSERT INTO** libri\_autori(id\_libro, id\_autore) VALUES (3, 2);

**INSERT INTO** libri\_generi(id\_libro, id\_genere) VALUES (3, 1);

**INSERT INTO** libri(id, ISBN, titolo, anno\_pubblicazione, quantita, id\_editore, id\_collana, id\_espositore)

VALUES (4, '4444444444444', 'La camera dei segreti', 1999, 8, 2, 2, 2);

**INSERT INTO** libri\_autori(id\_libro, id\_autore) VALUES (4, 2);

**INSERT INTO** libri\_generi(id\_libro, id\_genere) VALUES (4, 1);

**INSERT INTO** prestiti(id, id\_libro, id\_personale, id\_utente, data\_inizio, data\_fine) VALUES (1, 1, 2, 1, ‘2018-01-16’, ‘2018-02-15’);

**INSERT INTO** prestiti(id, id\_libro, id\_personale, id\_utente, data\_inizio, data\_fine) VALUES (2, 2, 2, 1, ‘2018-01-16’, ‘2018-02-15’);

**INSERT INTO** prestiti(id, id\_libro, id\_personale, id\_utente, data\_inizio, data\_fine) VALUES (3, 4, 1, 2, ‘2018-01-14’, ‘2018-02-13’);

## 4c) Query UPDATE

//Restituzione libro

**UPDATE** prestiti SET restituito=1 WHERE id=3;

## 4d) Query SELECT

//Ricerca prestiti per utenti

**SELECT** prestiti.id AS pres\_id, libri.titolo AS titolo, CONCAT(utenti.cognome, ' ',utenti.nome) AS utente, CONCAT(personale.cognome, ' ', personale.nome) AS person, data\_inizio, data\_fine, restituito

FROM prestiti

JOIN libri ON libri.id=prestiti.id\_libro

JOIN utenti ON utenti.id=prestiti.id\_utente

JOIN personale ON personale.id=prestiti.id\_personale

WHERE prestiti.id\_utente=1;

//Ricerca per autore

**SELECT** libri.id AS lib\_id,ISBN,titolo, anno\_pubblicazione, quantita, collane.nome AS collana, casa\_editrice.nome AS casa\_editrice, generi.nome AS genere, stanze.nome AS stanza, espositori.nome AS espositore

FROM libri

JOIN collane ON libri.id\_collana=collane.id

JOIN casa\_editrice ON libri.id\_editore=casa\_editrice.id

JOIN libri\_autori ON libri.id=libri\_autori.id\_libro

JOIN libri\_generi ON libri.id=libri\_generi.id\_libro

JOIN generi ON libri\_generi.id\_genere=generi.id

JOIN espositori ON espositori.id=libri.id\_espositore

JOIN stanze ON stanze.id=espositori.id\_stanza

WHERE libri\_autori.id\_autore=1;

//Ricerca per genere

**SELECT** libri.id AS lib\_id,ISBN,titolo, anno\_pubblicazione, quantita, collane.nome AS collana, casa\_editrice.nome AS casa\_editrice, CONCAT(autori.cognome, ' ', autori.nome) AS autore, stanze.nome AS stanza, espositori.nome AS espositore

FROM libri

JOIN collane ON libri.id\_collana=collane.id

JOIN casa\_editrice ON libri.id\_editore=casa\_editrice.id

JOIN libri\_autori ON libri.id=libri\_autori.id\_libro

JOIN libri\_generi ON libri.id=libri\_generi.id\_libro

JOIN autori ON libri\_autori.id\_autore=autori.id

JOIN espositori ON espositori.id=libri.id\_espositore

JOIN stanze ON stanze.id=espositori.id\_stanza

WHERE libri\_generi.id\_genere=1;

//Prestiti non restituiti oltre la data di scadenza

**SELECT** libri.titolo AS titolo, CONCAT(utenti.cognome, ' ', utenti.nome) AS utente, data\_inizio, data\_fine

FROM prestiti

JOIN libri ON prestiti.id\_libro=libri.id

JOIN utenti ON prestiti.id\_utente=utenti.id

WHERE restituito=0 AND data\_fine<'2018-02-30';

//Prestiti in un intervallo di tempo

**SELECT** libri.titolo AS titolo, CONCAT(utenti.cognome, ' ', utenti.nome) AS utente, CONCAT(personale.cognome, ' ', personale.nome) AS personale, data\_inizio, data\_fine, restituito

FROM prestiti

JOIN libri ON prestiti.id\_libro=libri.id

JOIN utenti ON prestiti.id\_utente=utenti.id

JOIN personale ON prestiti.id\_personale=personale.id

WHERE data\_inizio>='2018-01-01' AND data\_inizio<='2018-02-30';

//Libri prestati da un determinato membro del personale

**SELECT** titolo, CONCAT(utenti.cognome, ' ', utenti.nome) AS utente, data\_inizio, data\_fine, restituito

FROM prestiti

JOIN libri ON prestiti.id\_libro=libri.id

JOIN utenti ON prestiti.id\_utente=utenti.id

WHERE id\_personale=1;

//Ricerca per titolo del libro

**SELECT** libri.id AS id, ISBN, libri.titolo AS titolo, CONCAT(autori.cognome, ' ', autori.nome) AS autore,collane.nome AS collana, casa\_editrice.nome AS casa\_editrice,generi.nome AS genere, quantita AS quantita, stanze.nome AS stanza, espositori.nome AS espositore

FROM libri

JOIN collane ON libri.id\_collana=collane.id

JOIN casa\_editrice ON libri.id\_editore=casa\_editrice.id

JOIN libri\_autori ON libri.id=libri\_autori.id\_libro

JOIN libri\_generi ON libri.id=libri\_generi.id\_libro

JOIN generi ON libri\_generi.id\_genere=generi.id

JOIN autori ON libri\_autori.id\_autore=autori.id

JOIN espositori ON espositori.id=libri.id\_espositore

JOIN stanze ON stanze.id=espositori.id\_stanza

where libri.titolo like '%comp%';

//Inventario dei libri in biblioteca

**SELECT** collane.nome AS collana, titolo, quantita, quantita-COUNT(prestiti.id\_libro) AS disponibili

FROM prestiti

JOIN libri ON prestiti.id\_libro=libri.id

JOIN collane ON collane.id=libri.id\_collana

WHERE restituito=0

GROUP BY collana, titolo, quantita;