

## CATALOGAZIONE LIBRI BIBLIOTECA

1. Isolamento entità e raggruppamento proprietà;
  2. Isolare requisiti funzionali;
  3. Progettazione: schema logico con diagramma EER;
  4. Implementazione schema fisico e popolamento con i dati fake;
  5. Realizzazione delle viste;
  6. Realizzazione set tool con procedure e function;
- Presentazione lavoro.

## DESCRIZIONE

Censimento e inserire libri delle biblioteche statali toscane con informazioni dettagliate su tutti i libri presenti all'interno delle stesse. Queste informazioni devono includere il titolo, l'autore, l'ISBN, la categoria, il numero di copie disponibili e la posizione esatta dei libri all'interno delle biblioteche. Inoltre i libri devono essere categorizzati in base alla tipologia (storici, narrativa, saggistica e altre tipologie rilevanti). Il sistema di ricerca deve permettere di trovare i libri sia tramite ISBN che tramite titolo.

Un'altra funzionalità importante è la ricerca full text, che consente agli utenti di effettuare ricerche dettagliate all'interno dell'abstract dei libri. Questo richiede la creazione di un indice full text, che permetta di cercare parole chiave all'interno del testo dei libri, facilitando così la ricerca di informazioni specifiche e approfondite.

## ATTIVITA' 1: Isolamento entità e raggruppamento proprietà;

### Libri:

- **ID** - BIGINT
- **ISBN** - VARCHAR (13)  
ISBN è l'acronimo di International Standard Book Number.  
E' lo standard internazionale del mondo del libro che permette di riconoscere in modo univoco attraverso un codice un libro o un prodotto creato per essere usato come libro
- **TITOLO** - VARCHAR (255)
- **AUTORE** - VARCHAR (255) - null
- **ID\_Genere** BIGINT  
genere del libro (storico, narrativa,...)
- **ABSTRACT** (trama) - mediumtext
- **ANNO PUBBLICAZIONE** - INT - null
- **FOREIGN KEY** (ID\_Genere) REFERENCES Genere(id)

### Biblioteche:

- **ID** - BIGINT
- **ISIL** - VARCHAR(9)
- **DENOMINAZIONE** - VARCHAR(255)
- **VIA** - VARCHAR(255)
- **CIVICO** - VARCHAR(4)
- **COMUNE** - VARCHAR(50)

### CollocazioneLibri:

- **ID** - BIGINT
- **IDLibri** - BIGINT
- **IDBiblioteca** - BIGINT
- **Scaffale** - VARCHAR (2)
- **Posizione** - VARCHAR (3)
- **NumeroCopie** - INT
- **FOREIGN KEY** (IDLibri) REFERENCES libri(id)
- **FOREIGN KEY** (IDBiblioteca) REFERENCES biblioteca(id)

### Genere:

- **ID** - BIGINT
- **DescrizioneGenere** - VARCHAR (150)

## ATTIVITA' 2: Isolare requisiti funzionali;

**Ricerca per ISBN, titolo e Id:** Implementare funzionalità di ricerca che permettano agli utenti di trovare libri tramite ISBN, titolo e id.

**Ricerca full text:** Implementare un indice full text per cercare parole chiave all'interno degli abstract dei libri.

**Gestione delle categorie:** Consentire la categorizzazione dei libri in base alle diverse tipologie per facilitare la navigazione e la ricerca.

**Inserimento di nuovi libri:** Consentire l'inserimento di un nuovo libro in biblioteca, controllo del numero di copie disponibili, se il libro è già presente, incrementare il numero di copie.

## ATTIVITA' 3: Progettazione: schema logico con diagramma EER;

[diagramma EER](#)

## ATTIVITA' 4: Implementazione schema fisico e popolamento con i dati fake;

```
CREATE DATABASE GestionaleBiblioteche
```

```
CREATE TABLE Libri (
```

```
    ID BIGINT AUTO_INCREMENT PRIMARY KEY,  
    ISBN VARCHAR(13) NOT NULL,  
    Titolo VARCHAR(255) NOT NULL,  
    Autore VARCHAR(255),  
    ID_Genere BIGINT,  
    Abstract MEDIUMTEXT,  
    Anno_Pubblicazione INT,  
    FOREIGN KEY (ID_Genere) REFERENCES Genere(ID)
```

```
);
```

```
CREATE TABLE Biblioteche (
```

```
    ID BIGINT AUTO_INCREMENT PRIMARY KEY,  
    ISIL VARCHAR(9) NOT NULL,  
    Denominazione VARCHAR(255) NOT NULL,  
    Via VARCHAR(255) NOT NULL,  
    Civico VARCHAR(4) NOT NULL,  
    Comune VARCHAR(50) NOT NULL
```

```
);
```

```
CREATE TABLE CollocazioneLibri (
```

```
    ID BIGINT AUTO_INCREMENT PRIMARY KEY,  
    IDLibri BIGINT,  
    IDBiblioteca BIGINT,  
    Scaffale VARCHAR(2),  
    Posizione VARCHAR(3),
```

```
NumeroCopie INT,  
FOREIGN KEY (IDLibri) REFERENCES Libri(ID),  
FOREIGN KEY (IDBiblioteca) REFERENCES Biblioteche(ID));
```

```
CREATE TABLE Genere (  
    ID BIGINT AUTO_INCREMENT PRIMARY KEY,  
    DescrizioneGenere VARCHAR(150) NOT NULL  
);
```

```
INSERT INTO Libri (ISBN, Titolo, Autore, ID_Genere, Abstract, Anno_Pubblicazione)  
VALUES  
(9781234567897, 'Il Nome della Rosa', 'Umberto Eco', 1, 'Un giallo storico ambientato in un  
monastero benedettino.', 1980),  
(9781234567898, '1984', 'George Orwell', 4, 'Una distopia futuristica sul totalitarismo.',  
1949),  
(9781234567899, 'Il Signore degli Anelli', 'J.R.R. Tolkien', 4, 'Un classico della narrativa  
fantasy.', 1954);
```

```
INSERT INTO Genere (DescrizioneGenere) VALUES  
(Storico),  
(Narrativa),  
(Saggistica),  
(Fantascienza),  
(Giallo),  
(Biografia);
```

```
INSERT INTO Biblioteche (ISIL, Denominazione, Via, Civico, Comune) VALUES  
(IT-FI001, 'Biblioteca Nazionale Centrale di Firenze', 'Piazza dei Cavalleggeri', '1', 'Firenze'),  
(IT-PI001, 'Biblioteca Universitaria di Pisa', 'Lungarno Pacinotti', '17', 'Pisa'),  
(IT-SI001, 'Biblioteca Comunale degli Intronati', 'Via della Sapienza', '5', 'Siena');
```

```
INSERT INTO CollocazioneLibri (IDLibri, IDBiblioteca, Scaffale, Posizione, NumeroCopie)  
VALUES  
(1, 1, 'A1', '001', 3),  
(2, 2, 'B2', '045', 5),  
(3, 3, 'C3', '078', 2);
```

```
ALTER TABLE Libri ADD FULLTEXT (Abstract);
```

```
use gestionale_biblioteche;
```

```
CREATE TABLE CollocazioneLibri (  
    ID BIGINT AUTO_INCREMENT PRIMARY KEY,  
    IDLibri BIGINT,  
    IDBiblioteca BIGINT,  
    Scaffale VARCHAR(2),  
    Posizione VARCHAR(3),
```

```

    NumeroCopie INT,
    FOREIGN KEY (IDLibri) REFERENCES Libri(ID),
    FOREIGN KEY (IDBiblioteca) REFERENCES Biblioteche(ID)
);
INSERT INTO CollocazioneLibri (IDLibri, IDBiblioteca, Scaffale, Posizione, NumeroCopie)
VALUES
(1, 1, 'A1', '001', 3),
(2, 2, 'B2', '045', 5),
(3, 3, 'C3', '078', 2);

```

## ATTIVITA' 5: Realizzazione delle viste;

### VIEW PER CREARE UNA LISTA DI TUTTI I DETTAGLI PER I LIBRI PRESENTI NELLE BIBLIOTECHE

```

CREATE VIEW dettagliLibri as select L.Id, L.Titolo, L.Autore, L.ISBN, G.DescrizioneGenere
as Genere, L.Abstract as Trama, concat(B.Denominazione,' ', B.Via, ' ', B.Civico, ' ',
B.Comune) as Biblioteca ,CL.scaffale, CL.posizione, CL.NumeroCopie
from collocazionelibri CL
inner join libri L on CL.IDLibri = L.ID
inner join biblioteche B on B.Id = CL.IdBiblioteca
inner join genere G on G.Id = L.Id_Genere;

```

## ATTIVITA' 6: Realizzazione set tool con procedure e function;

### PROCEDURA PER RICERCARE I LIBRI ATTRAVERSO IL CODICE ISBN

```

CREATE PROCEDURE `Ricerca_ISBN`(ISBN VARCHAR(13))
BEGIN
    select * from dettagliLibri where ISBN = dettaglilibri.ISBN;
END

```

### PROCEDURA PER RICERCARE I LIBRI ATTRAVERSO IL TITOLO

```

CREATE PROCEDURE `ricerca_per_titolo`(titolo VARCHAR(255))
BEGIN
select * from dettagliLibri where titolo = dettaglilibri.titolo;
END

```

### PROCEDURA PER RICERCARE I LIBRI ATTRAVERSO L'ID\_LIBRI

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `Ricerca_ID`(idLibri bigint)
BEGIN
    select * from dettagliLibri where idLibri = dettaglilibri.Id;
END

```

PROCEDURA PER RICERCARE I LIBRI ATTRAVERSO IL GENERE

```
CREATE PROCEDURE `Ricerca_Genere`(genere varchar(150))
BEGIN
select * from dettagliLibri where genere = dettagliLibri.Genere;
END
```

PROCEDURA PER RICERCARE I LIBRI ATTRAVERSO DELLE PAROLE CHE SONO CONTENUTE NELLA TRAMA DEL LIBRO

```
CREATE PROCEDURE `Ricerca_Trama`(parole mediumtext)
BEGIN
select * from dettagliLibri DL where MATCH(DL.Trama) AGAINST (parole);
END
```

PROCEDURA PER INSERIRE LIBRO E CONTROLLARE SE IL LIBRO È GIÀ PRESENTE, NEL CASO IN CUI È PRESENTE INCREMENTARE IL NUMERO COPIE IN

DELIMITER //

```
CREATE PROCEDURE InserisciLibro(
  IN p_ISBN VARCHAR(13),
  IN p_Titolo VARCHAR(255),
  IN p_Autore VARCHAR(255),
  IN p_DescrizioneGenere VARCHAR(150),
  IN p_Abstract MEDIUMTEXT,
  IN p_Anno_Pubblicazione INT,
  IN p_IDBiblioteca BIGINT,
  IN p_Scaffale VARCHAR(2),
  IN p_Posizione VARCHAR(3),
  IN p_NumeroCopie INT
)
BEGIN
  DECLARE v_ID_Genere BIGINT;
  DECLARE v_ID_Libro BIGINT;

  -- Controlla se il genere esiste, altrimenti aggiungilo
  SELECT ID INTO v_ID_Genere
  FROM Genere
  WHERE DescrizioneGenere = p_DescrizioneGenere;

  IF v_ID_Genere IS NULL THEN
    INSERT INTO Genere (DescrizioneGenere) VALUES (p_DescrizioneGenere);
    SET v_ID_Genere = LAST_INSERT_ID();
  END IF;

  -- Controlla se il libro esiste già
```

```
SELECT ID INTO v_ID_Libro
FROM Libri
WHERE ISBN = p_ISBN;
```

```
IF v_ID_Libro IS NULL THEN
```

**-- Inserisci il nuovo libro**

```
INSERT INTO Libri (ISBN, Titolo, Autore, ID_Genere, Abstract, Anno_Pubblicazione)
VALUES (p_ISBN, p_Titolo, p_Autore, v_ID_Genere, p_Abstract,
p_Anno_Pubblicazione);
SET v_ID_Libro = LAST_INSERT_ID();
```

**-- Inserisci la collocazione del libro**

```
INSERT INTO CollocazioneLibri (IDLibri, IDBiblioteca, Scaffale, Posizione,
NumeroCopie)
VALUES (v_ID_Libro, p_IDBiblioteca, p_Scaffale, p_Posizione, p_NumeroCopie);
ELSE
```

**-- Incrementa il numero di copie se il libro esiste già**

```
UPDATE CollocazioneLibri
SET NumeroCopie = NumeroCopie + p_NumeroCopie
WHERE IDLibri = v_ID_Libro AND IDBiblioteca = p_IDBiblioteca;
```

**-- Se non esiste una collocazione per questo libro e biblioteca, aggiungila**

```
IF ROW_COUNT() = 0 THEN
    INSERT INTO CollocazioneLibri (IDLibri, IDBiblioteca, Scaffale, Posizione,
NumeroCopie)
    VALUES (v_ID_Libro, p_IDBiblioteca, p_Scaffale, p_Posizione, p_NumeroCopie);
END IF;
END IF;
END //
```

```
DELIMITER ;
```