

# Schema Relazionale

User(id, username, password, nome, cognome, email)

Anagrafica(id, id\_Riconoscimento, nome, cognome, email)

Libro(id, ISBN, titolo, autore, sezione, scaffale)

Prestito(id, idLibro, idAnagrafica, dataInizio, scadenza, dataRiconsegna)

Prestito.idLibro -> Libro.id

Prestito.idAnagrafica -> Anagrafica.id

## Analisi Schema Database

Lo schema proposto risolve i principali problemi richiesti all'interno del testo della prova.

L'entità **User** rappresenta la tabella che contiene l'elenco dei bibliotecari e le informazioni necessarie per il login al gestionale (username e password).

In questo caso è rappresentata come una entità a se stante, perché ho previsto l'utilizzo del gestionale da parte di un solo bibliotecario.

Nel caso in cui ci siano più bibliotecari, potrebbe essere utile collegare l'entità alla relazione prestito, per tenere traccia del bibliotecario che lo ha effettuato, così da poter risalire a quest'ultimo. Si potrebbe addirittura pensare di unire le due entità (Anagrafica e User) poiché differiscono solo per il ruolo, quindi avere una singola entità che contiene l'unione degli attributi di entrambe.

Nella tabella **Anagrafica**, una volta registrato, troveremo tutte le informazioni necessarie all'individuazione di uno studente, e la sua email per eventuali comunicazioni, come ad esempio l'imminente scadenza di un prestito.

Per supportare la necessità di poter effettuare prestiti a persone che non siano studenti, una possibile soluzione potrebbe essere l'inserimento di un "flag" (studente) come attributo della tabella anagrafica, che sarà di tipo boolean con valore di default a TRUE, poiché si tratta di una biblioteca universitaria.

Inoltre, mentre per gli studenti sarà possibile utilizzare il numero di matricola e il numero di carta di identità come documento di riconoscimento, per utenti esterni all'università, sarà obbligatorio utilizzare il numero di carta d'identità.

Quindi per distinguere tra studenti e non studenti che hanno utilizzato la carta d'identità come documento di riconoscimento ci viene in aiuto il flag descritto prima.

Potrebbe anche essere possibile utilizzare il numero di un qualsiasi documento di riconoscimento, in tal caso al momento della registrazione potrei memorizzare per l'utente anche il tipo di documento utilizzato tra un numero finito di opzioni, cosicché il bibliotecario possa avere, al momento della ricerca, anche quale documento di riconoscimento richiedere per la verifica dell'identità del cliente.

Nella tabella **Libri** saranno salvate le informazioni di ciascun libro, di cui potranno essere inserite più copie, e le informazioni per poterlo trovare fisicamente all'interno della biblioteca.

In questo caso sono stati considerati solamente i campi sezione e scaffale per definire la posizione, dato che risulta logico pensare che libri uguali si trovino nello stesso scaffale.

Nella tabella **Prestito** saranno registrati tutti i prestiti, compresi quelli già conclusi.

Qui vedremo quindi quale utente ha quale libro e la data di inizio prestito, la data di scadenza del prestito e la data di riconsegna effettiva del libro, nel caso in cui si debbano prendere provvedimenti per un ritardo nella restituzione.

Nello schema proposto, ho previsto che tutti gli utenti della biblioteca alla fine riconsegnino il libro preso in prestito, se però dovessimo tenere traccia anche della mancata restituzione di un libro, sarà sufficiente aggiungere alla tabella Prestito un attributo (restituito) di tipo boolean che verrà lasciato FALSE.