



Università degli Studi di Ferrara

a.a. 2019-2020

Progetto Basi di Dati

Database Biblioteca

Marco Beltrame - Enrico Manfrin

Gruppo 13

September 17, 2020

Indice

1 Prefazione	1
2 Introduzione	2
3 Descrizione del Problema	3
4 Modello Concettuale	4
4.1 Regole di Lettura	10
5 Modello Logico	10
5.1 Normalizzazione	13
6 Modello Fisico	14
7 Algebra Relazionale - Linguaggio SQL	15
7.1 Statistiche di Base	15
7.2 Interrogazioni Aggiuntive	18
8 Pagine Web	22
8.1 Descrizione delle Pagine Web	23
9 Conclusione	26

1 Prefazione

Con questo progetto, in base alle nozioni acquisite durante il corso di Basi di Dati e Laboratorio dell'anno accademico 2019/2020, intendiamo risolvere nel migliore dei modi il problema commissionatoci dalla biblioteca universitaria dell'Università degli Studi di Ferrara.

Complessivamente tale lavoro ci ha sottratto tempo e fatica (mentale), tuttavia siamo convinti che ci abbia altresì ricompensati, facendoci ottenere una discreta base progettuale nell'ambito delle Basi di Dati e, siamo convinti che ci sia servito anche come “allenamento” per la preparazione di piani di lavoro futuri, ad esempio per la Tesi di Laurea.

Ci auguriamo di aver soddisfatto a pieno tutte le richieste assegnateci dai docenti.

Vogliamo inoltre scusarci qualora possiate riscontrare eventuali e leciti errori di battitura e/o grammaticali nel corso della correzione di questo elaborato.

Ci si perdoni anche per non aver sfruttato al meglio L^AT_EX, strumento quest'ultimo per noi ancora in fase di apprendimento. In siffatta tesina sono presenti alcune immagini che raffigurano le idee e i concetti soggiacenti ai nostri ragionamenti ed alle nostre ipotesi.

Per una miglior visualizzazione ai fini dell'analisi di tali figure, ogni qualvolta che le incontrerete, quello che vi consigliamo è di mantenere uno zoom di circa il 270% di questo documento (testato su un elaboratore con monitor a 15").

Buona lettura.

2 Introduzione

Il Progetto è stato realizzato da: Marco Beltrame (matricola 148881), Enrico Manfrin (matricola: 143480). Il gruppo di riferimento è il #13.

È stato portato a compimento per il 50% da entrambi gli autori. In prima istanza il progetto è stato **divertente** e ci ha permesso di ampliare le nostre conoscenze connesse al mondo dell'informatica. In particolare, abbiamo appreso come gestire una base di dati in tutte le sue fasi, a partire dall'individuazione di un certo aspetto del mondo reale, andando ad analizzare i requisiti richiesti dalla biblioteca universitaria, per passare in seguito alla progettazione del modello dei dati, avvenuta in tre livelli diversi:

- A livello concettuale, in cui abbiamo iniziato a modellare le singole entità (con i relativi attributi) ed associazioni tra esse, costituenti il Modello E/R. (Vedi [Modello E/R](#)).
- Derivando tale diagramma in Schema Relazionale siamo quindi passati al livello logico, la cui realizzazione ha visto l'applicazione di alcune regole di trasformazione viste a lezione. Tale modello è stato successivamente normalizzato in 3NF (fino alla terza forma normale). (Vedi [Modello Relazionale](#)).
- L'implementazione su disco del livello logico, grazie al DBMS MySQL Workbench ci ha permesso infine di raggiungere il livello fisico.

Dopo aver definito e popolato la nostra base di dati, l'ulteriore passo è stato quello di scrivere ed assicurarci della correttezza di tutte le interrogazioni della sezione “Statistiche di base” in SQL con l'equivalente espressione scritta in Algebra Relazionale (senza usare l'operatore di divisione). In aggiunta a queste, abbiamo ideato altre interrogazioni alla nostra base di dati. Si veda la [sezione corrispondente](#).

Infine, abbiamo avuto la possibilità di familiarizzare con alcuni strumenti visti durante il corso, atti alla gestione delle pagine web, tra cui: HTML, PHP e CSS. Oltre a questi, ci siamo serviti in minima parte di: JQuery, Ajax, Icons, Chart e di L^AT_EX, per la stesura di questa tesina.

È possibile vedere in maniera schematica l'organizzazione dei file rappresentanti le varie [pagine del sito](#), la quale è seguita da una [descrizione](#) del ruolo e del funzionamento di ciascuna.

3 Descrizione del Problema

È opportuno premettere che, per rappresentare e successivamente risolvere il problema non sono stati presi in considerazione i modelli: gerarchico, reticolare e ad oggetti; è invece stato adottato il modello relazionale, basato sul concetto di insieme e sulla strutturazione dei dati tramite tabelle. Questa prima fase del processo di sviluppo è detta **analisi dei requisiti** e riguarda la conoscenza degli obiettivi. Infatti, conoscere ciò che si dovrà sviluppare è il fattore di successo indispensabile per operare nella giusta direzione. Nell'ambito del progetto informatico è emersa evidente per noi “progettisti” la necessità di approfondire la conoscenza dei fattori rilevanti per un corretto risultato, oltre a possedere la necessaria professionalità informatica. Per ottenere tale conoscenza è indispensabile poter disporre di esperti della materia di pertinenza del progetto.

L'**intervista** è una tecnica, utilizzata nello sviluppo di progetti informatici, che ha come obiettivo quello di conoscere, comprendere e documentare la materia oggetto del progetto: si basa sul metodo dell’indagine conoscitiva. L’intervista non sviluppa il progetto: essa consente di acquisire il dominio della materia. Non è possibile progettare senza la padronanza culturale di quanto è oggetto del progetto. Il progettista (noi) che ha l’incarico di sviluppare un progetto informatico, utilizza la tecnica dell’intervista per scoprire tutte le componenti e le caratteristiche del suo progetto. Per scoprire ciò, deve attingere le informazioni dalle **fonti** che detengono la conoscenza della materia. Esse sono i testi, le persone esperte, coloro che svolgono mansioni inerenti l’area del progetto. La nostra fonte è stata un **testo**, reso disponibile dalla biblioteca universitaria dell’Università degli Studi di Ferrara. Tralasciamo, per motivi di tempo e spazio gli altri aspetti caratterizzanti la progettazione, che discendono da quanto appena raccontato, ma che esulano da quelli affrontati durante il corso. Capiamo ora il **problema**.

Era richiesto di realizzare un’applicazione web scritta in un linguaggio lato server, come il php che facesse uso di un database per modellare e gestire la biblioteca menzionata poc’anzi in modo da memorizzare le informazioni relative a *libri, utenti e prestiti*.

Abbiamo recuperato i “dati (grezzi)” da 3 diverse fonti forniteci dal nostro *cliente* (che, ricordiamo essere la biblioteca). Tramite quindi: un **link** che porta alla pagina web dei dipartimenti con le varie sedi; una **tabella** in fondo al testo che rappresentava gli editori dei libri; un **file CSV** “Dati.csv” contenente le informazioni sui libri, sulle copie ed autori di questi. Al contrario, ci siamo inventati i dati relativi agli unici utenti della biblioteca, ossia gli studenti.

Nella sezione successiva discuteremo il modo in cui abbiamo organizzato questi “dati grezzi”.

Inserimento, modifica e cancellazione sono 3 delle operazioni richieste riguardanti sia gli utenti che i prestiti; per questi ultimi, si desidera visualizzarne anche la situazione (ossia quali presiti sono in corso). Erano inoltre richieste operazioni per effettuare la ricerca di un utente o di un libro (inserendo anche solo parzialmente il nome o il titolo) per ricavare gli elenchi di: tutti i libri dati in prestito ad un utente, tutti gli utenti che hanno in prestito un determinato libro ed alcune **statistiche basiliari**:

1. Determinare le 5 lingue più comuni in cui sono scritti i libri presenti nella basi di dati.
2. Determinare l'autore che ha scritto più libri.
3. Determinare l'editore che ha pubblicato più libri.

In modo del tutto facoltativo, erano ben gradite altre **interrogazioni aggiuntive**, lasciate alla creatività del progettista. Ulteriori precisazioni e vincoli del Minimondo derivanti dal testo sono:

- La maggior parte dei titoli dei libri sono generati casualmente.
- Si esclude dall'elenco la Facoltà di Medicina, Farmacia e Prevenzione anche se presente nella pagina web dei dipartimenti poichè non ha una sede specificata.
- L'applicazione web non deve avere il login; si suppone che l'indirizzo web (url) di accesso sia accessibile solamente dal personale autorizzato
- Al momento non ci sono presenti libri scritti in più lingue, ogni libro è presente in una sola lingua anche se in più copie, quindi l'ISBN è lo stesso per tutte le lingue
- Un libro può essere pubblicato da una sola casa editrice.

4 Modello Concettuale

La progettazione dello schema concettuale ha avuto un ruolo importantissimo per la vita del nostro progetto. Abbiamo affrontato il problema seguendo alcuni criteri.

Prima di tutto abbiamo letto più volte ed attentamente il testo prima di decidere il modello dei dati. Ovviamente, abbiamo escluso l'ambito del problema dall'insieme delle possibili entità: infatti, nella prima riga della Traccia del Progetto - “La biblioteca universitaria dell’Università degli Studi di Ferrara vi ha commissionato la realizzazione...” - la biblioteca, o meglio il sistema bibliotecario in sè non è un’entità, bensì costituisce il nostro *Minimondo*.

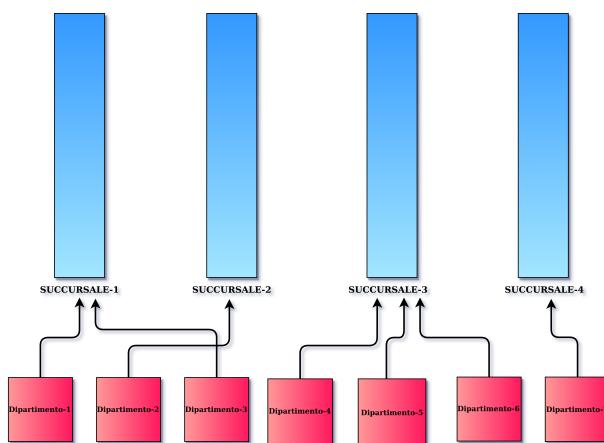
Abbiamo dunque deciso che, i sostantivi che incontravamo nel testo e che talvolta occorrevano in

modo sparso nella traccia, erano dei buoni candidati per diventare entità, mentre, gli aggettivi e le proprietà che arricchivano questi nomi sarebbero corrisposti agli attributi; infine, i verbi alle associazioni. In aggiunta, ci siamo soffermati anche nelle richieste di output e sul testo delle interrogazioni, per essere sicuri di aver determinato correttamente e in modo completo gli attributi delle entità. Durante tutto il processo decisionale di progettazione, abbiamo avuto diverse idee, a volte contrastanti su come gestire un certo elemento del modello. Abbiamo convenuto però in un' **importante ipotesi** che ora accuriamo insieme:

La biblioteca universitaria dell'Università degli Studi di Ferrara non è fisicamente presente in un unico e determinato luogo, ma è suddivisa lungo la città di Ferrara in più sedi, cioè in più succursali. Ogni succursale ospita, ovvero accomuna uno o più dipartimenti:

Si pensi ad esempio alla sede presente al Polo Scientifico Tecnologico in Via Giuseppe Saragat 1, la quale offre il servizio di gestione di prestiti dei libri a studenti provenienti da 2 diversi Dipartimenti, quali: Ingegneria e Fisica e Scienze della Terra (nella realtà anche gli studenti di Informatica possono accedervi, ma escludiamo questi ultimi dall'elenco ai fini del modello dei dati che stiamo considerando, in quanto, il nome della via in cui è ubicato il Dipartimento di Matematica ed Informatica non è lo stesso della Succursale).

SUDDIVISIONE BIBLIOTECA IN SUCCURSALI



La Succursale 1 eroga il servizio di prestiti a studenti che frequentano i Dipartimenti 1 e 2.
 La Succursale 2 eroga il servizio di prestiti a studenti che frequentano il Dipartimento 2
 La Succursale 3 eroga il servizio di prestiti a studenti che frequentano i Dipartimenti 4, 5 e 6
 La Succursale 4 eroga il servizio di prestiti a studenti che frequentano il Dipartimento 7

Tra l'altro, anticipiamo il fatto che, dall'immagine si evincono i vincoli di cardinalità massima e quelli di dipendenza di esistenza (minime) tra le “entità” SUCCURSALE — DIPARTIMENTO.

Appurato ciò, gli oggetti ai quali abbiamo dato dignità di esistere come *entità* sono:

LIBRO, AUTORE, EDITORE, COPIA, STUDENTE, SUCCURSALE, DIPARTIMENTO.

Di un libro possono esistere o meno copie. Se ne sussistono, allora possiamo averne 1 o più afferenti a quel determinato libro.

L'entità **COPIA** è stata modellata come *entità debole*, poichè fa riferimento ad un determinato libro, dunque, in quanto replica avrà lo stesso titolo e lo stesso ISBN di quel libro; a quest'ultima abbiamo deciso di aggiungere un attributo 'Cod_Copia' di tipo intero progressivo per identificare il numero di copia di quel libro (ossia quale copia stiamo considerando tra quelle appartenenti all'insieme di tutte le copie di un determinato libro). 'Cod_Copia' è una "chiave debole". Questo risponde alla specifica della traccia: - "... A ciascun libro è inoltre associato un codice univoco (l'ISBN non è infatti univoco per i libri che esistono in più copie) ..." -.

Le entità **EDITORE**, **AUTORE** e **STUDENTE** seppure ricche di contenuto informativo, non presentano particolarità; infatti per quest'ultime non abbiamo implementato ulteriori specifiche oltre a quelle standard del testo (non abbiamo quindi inserito attributi aggiuntivi). Sappiamo dalla traccia del problema che ad ogni libro dev'essere associato un solo editore e che gli utenti della biblioteca sono esclusivamente studenti.

Sottolineiamo che, quando uno studente desidera prendere in prestito un determinato libro, ciò che effettivamente gli viene prestato è una copia di quel libro, e non il libro in sè.

Sappiamo anche che la lingua in cui è scritto un libro può essere una qualsiasi, il titolo del libro invece sarà sempre in lingua inglese (con affiancata eventualmente la lingua originale) al fine di facilitare la ricerca, come da consegna. Abbiamo quindi inserito l'attributo 'Lingua' nell'entità **LIBRO**. Le informazioni e i dati relativi ad un prestito sono rappresentati dall'associazione '**Prestito**' e dai suoi due attributi 'Data_Uscita', 'Data_Limite'.

Continuando ad analizzare il nostro percorso progettuale, troviamo l'entità **SUCCURSALE** che caratterizza una delle Succursali che, nel loro insieme costituiscono l'intera Biblioteca Ferrarese; in altre parole, essa rappresenta una delle varie sedi, una "filiale" tra tutte quelle che formano il Polo Bibliotecario Ferrarese in cui gli studenti prendono in prestito le copie dei libri.

A questa, per esigenze implementative abbiamo aggiunto l'attributo 'Cod_Suc'. **SUCCURSALE** è in collegamento con **COPIA**. Tale associazione si chiama '**Conservare**'. Siamo ora giunti ad una **seconda importante scelta**:

Siamo a conoscenza del fatto che: ogni libro esiste in più copie che possono essere fisicamente conservate in succursali diverse, infatti ogni dipartimento dell'Università ha la propria biblioteca

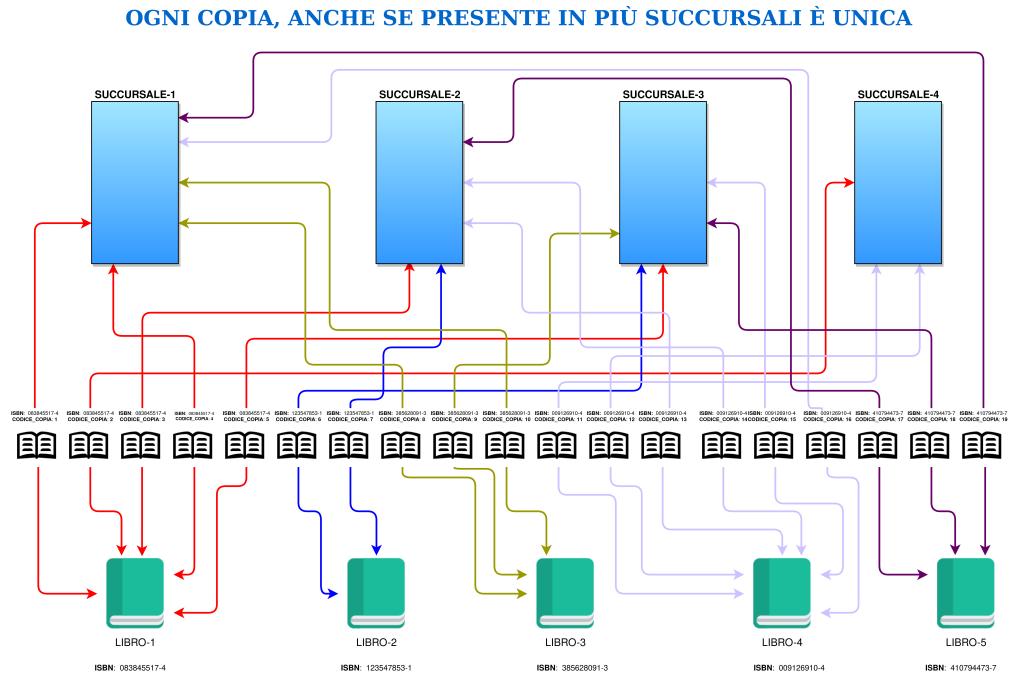
(succursale) ed un libro potrebbe essere presente in diverse sedi.

Tutte le copie di un determinato libro ereditano lo stesso ISBN; ci serve quindi un modo per contraddistinguere in maniera univoca le varie copie che può possedere suddetto libro.

L'attributo ‘*Cod_Copia*’ presente in COPIA, ci permette di distinguere quella specifica copia dalle altre; questo fatto ci porta ad asserire che, quella particolare copia col suo codice (che la identifica univocamente) è presente in modo unico.

Grazie al connubio tra ‘*Cod_Copia*’ ed ISBN, possiamo differenziare quella copia dall’insieme di tutte le copie presenti nella base di dati, ossia da tutte quelle che sono conservate all’interno delle varie succursali (di alcune, poichè non è detto che ci sia per forza in ogni succursale).

Consideriamo ad esempio un caso limite per cui in ogni succursale S_1, S_2, \dots, S_N siano conservate più copie C_1, C_2, \dots, C_M di uno stesso libro L. Sebbene siano tutte uguali in quanto replice di L, ogni singola copia $C_i, i=0, \dots, M$, è unica all’interno di ognuna delle N succursali.



In modo del tutto analogo a prima, da questa figura possiamo capire anche la nostra scelta circa le cardinalità tra le entità LIBRO — COPIA e tra COPIA — SUCCURSALE. Per cercare il più possibile di non lasciare dubbi al lettore sul nostro modello, riteniamo opportuno disambiguare anche l’associazione molti a molti ‘**Prestito**’ che connette le entità ‘COPIA’ e ‘STUDENTE’.

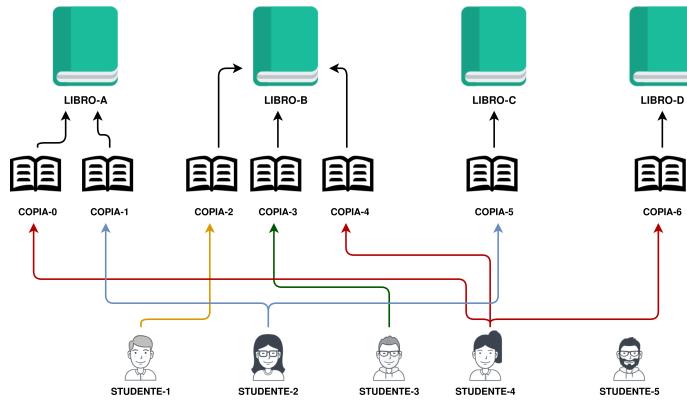
Se esistono copie di un libro, allora una o più di esse sono date in prestito a zero o più studenti.

0 o più studenti perché, in un dato momento, potremmo avere uno o più studenti che non hanno in

prestito alcuna copia, mentre altri che ne possiedono una o più. Gli studenti privi di copie, avranno la facoltà di domandarne in prestito, in momenti successivi. Detta cardinalità esprime quindi la “discrezione” degli studenti nell’ottenere in prestito, o meno, una o più copie.

Riassumendo, quindi: in ogni momento 0 o più copie sono in prestito a 0 o più studenti, d’altra parte, 0 o più studenti prendono in prestito 0 o più copie.

SISTEMA DI PRESTITI DELLE COPIE AGLI STUDENTI



Per ultima abbiamo l’entità ‘**DIPARTIMENTO**’ anch’essa intuitiva. L’associazione ‘*Ospitare*’ tra ‘SUCCURSALE’ e ‘DIPARTIMENTO’ è conforme all’ipotesi [di cui sopra](#). È evidente che, durante la costruzione dello schema, abbiamo deciso anche un insieme minimale di attributi che ci permettessero di distinguere tra loro le istanze di una stessa entità, ossia le **chiavi primarie**:

- EDITORE: Cod_E
- LIBRO: ISBN
- AUTORE: Cod_A
- COPIA: Cod_Copia, ISBN
- STUDENTE: Matricola
- SUCCURSALE: Cod_Suc
- DIPARTIMENTO: Nome_D

Inoltre, per ogni associazione, abbiamo dissertato su quali entità vi partecipassero in modo facoltativo (0) e quali in modo obbligatorio (1 o N). Fatto ciò, abbiamo deciso le cardinalità delle associazioni.

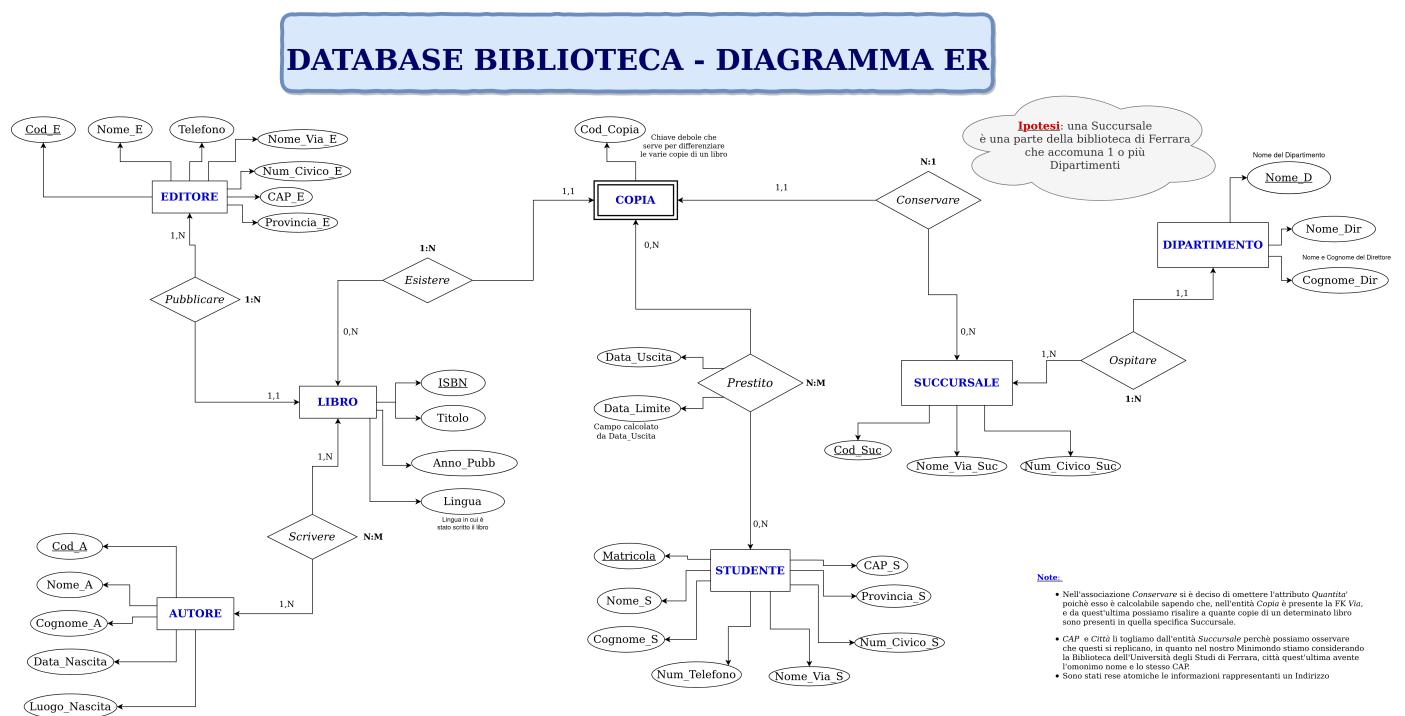


Figure 1: Modello E/R

4.1 Regole di Lettura

Leggendo da sinistra verso destra, lo schema¹ può essere così interpretato:

- Un editore *pubblica* uno o più libri; ogni libro è *pubblicato* da uno ed un solo editore.
- Uno o più autori *scrivono* uno o più libri; un libro è *scritto* da uno o più autori.
- Un libro può *esistere* in più copie; ogni copia *esistente* fa riferimento ad uno ed un solo libro.
- Ogni studente può *prendere in prestito* zero o più copie; ogni copia è *prestata* a zero o più studenti.
- Ogni copia è *conservata* in una ed una sola succursale; in ogni succursale sono *conservate* zero o più copie.
- Ogni succursale *ospita* uno o più dipartimenti; ogni dipartimento è *ospitato* da una ed una sola succursale.

5 Modello Logico

In questo modello una base di dati è vista (informalmente) come un insieme di tabelle sulle quali possono essere eseguite opportune operazioni. Si chiama così perchè è basato sul concetto matematico di relazione tra insiemi di oggetti. Numerosi sono gli indispensabili concetti che abbiamo studiato per questa fase del progetto, alcuni di essi sono: **schema/intensione della relazione R** definito sugli attributi $A_i, i=1,\dots,N$; **estensione/stato** della relazione ($r(R)$); **tuple** (righe della tabella); **dominio** di un attributo; $t[A_i] = v_i$ (valore dell'attributo A_i per la tupla t); **vincoli di integrità** (sulla chiave, dell'entità, referenziale) ecc. Come preannunciato nell'introduzione, abbiamo seguito l'algoritmo di mappatura da schema ER a Relazionale; ricordiamo brevemente al lettore il suo funzionamento:

1. ogni *entità* diventa una relazione;
2. ogni *attributo* di un'entità diventa un attributo della relazione, cioè il nome di una colonna della tabella;
3. ogni *attributo* della relazione eredita le caratteristiche dell'attributo dell'entità da cui deriva;
4. l'identificatore univoco di un'entità diventa la *chiave primaria* della relazione derivata;

¹Per una miglior visualizzazione dello schema con il software Draw.IO si visiti la relativa [Pagina](#).

5. l'associazione *uno a uno* diventa un'unica relazione che contiene gli attributi della prima e della seconda entità. NB: esistono altri 2 modi per gestire questo tipo di legame fra entità;
6. l'associazione *uno a molti* viene rappresentata aggiungendo, agli attributi dell'entità che svolge il ruolo a molti, l'identificatore univoco dell'entità che svolge il ruolo a uno nell'associazione. Questo identificatore, che prende il nome di chiave esterna (*foreign key*) dell'entità associata, è costituito dall'insieme di attributi che compongono la chiave dell'entità a uno dell'associazione. Gli eventuali attributi dell'associazione vengono inseriti nella relazione che rappresenta l'entità a molti, assieme alla chiave esterna.
7. L'associazione *molti a molti* diventa una nuova relazione (in aggiunta alle relazioni derivate dalle entità) composta dagli identifieri univoci delle due entità e dagli eventuali attributi dell'associazione. La chiave della nuova relazione è formata dall'insieme di attributi che compongono le chiavi delle due entità, oltre agli attributi dell'associazione necessari a garantire l'unicità delle tuple nella relazione ottenuta.

Esaminando lo schema relazionale riportato a pagina successiva, notiamo che per l'*entità debole* COPIA è stata creata la corrispondente relazione; la sua chiave primaria è la combinazione tra la PK dell'entità proprietaria LIBRO e della sua chiave parziale. Abbiamo unito le chiavi delle relazioni con una freccia: FK → PK. Notiamo dall'ER che non sono presenti associazioni **1:1**. Invece, possiamo rilevare ben 4 associazioni **1:N** tra: EDITORE-LIBRO, LIBRO-COPIA, SUCCURSALE-COPIA, SUCCURSALE-DIPARTIMENTO. Ciò implica che, agli attributi della relazione possedente il lato 'a molti' sia aggiunta la chiave dell'entità 'a uno'. Per esempio, tra EDITORE (lato 1) e LIBRO (lato N), notiamo che in fondo allo schema della relazione LIBRO è stato aggiunto l'attributo 'C_E' avente ruolo di chiave esterna, il quale punta, tramite un arco diretto, alla chiave primaria 'COD_E' di EDITORE. Similmente, l'attributo FK 'ISBN_LIBRO' della relazione referenziante COPIA, fa riferimento all'attributo PK 'ISBN' della relazione riferita LIBRO. Osserviamo come 'ISBN_LIBRO', in combinazione con 'COD_COPIA' funga anche da PK, oltre ad essere una chiave esterna. Con il medesimo meccanismo si sono realizzati gli altri due schemi delle relazioni uno a molti. Di associazioni **N:M** abbiamo '*Scrivere*' e '*Prestito*'. Queste sono state tradotte nelle due relazioni SCRIVERE e PRESTITO. Entrambe contengono come chiavi esterne le PK delle due relazioni e gli eventuali attributi dell'associazione. Infatti, nello schema di PRESTITO troviamo 'ISBN_LIB', 'C_COPIA', 'MATR' come chiave esterna (e anche come PK) e 'DATA_USCITA', 'DATA_LIMITE' come attributi ereditati dell'associazione. Lo stesso si applica a SCRIVERE.

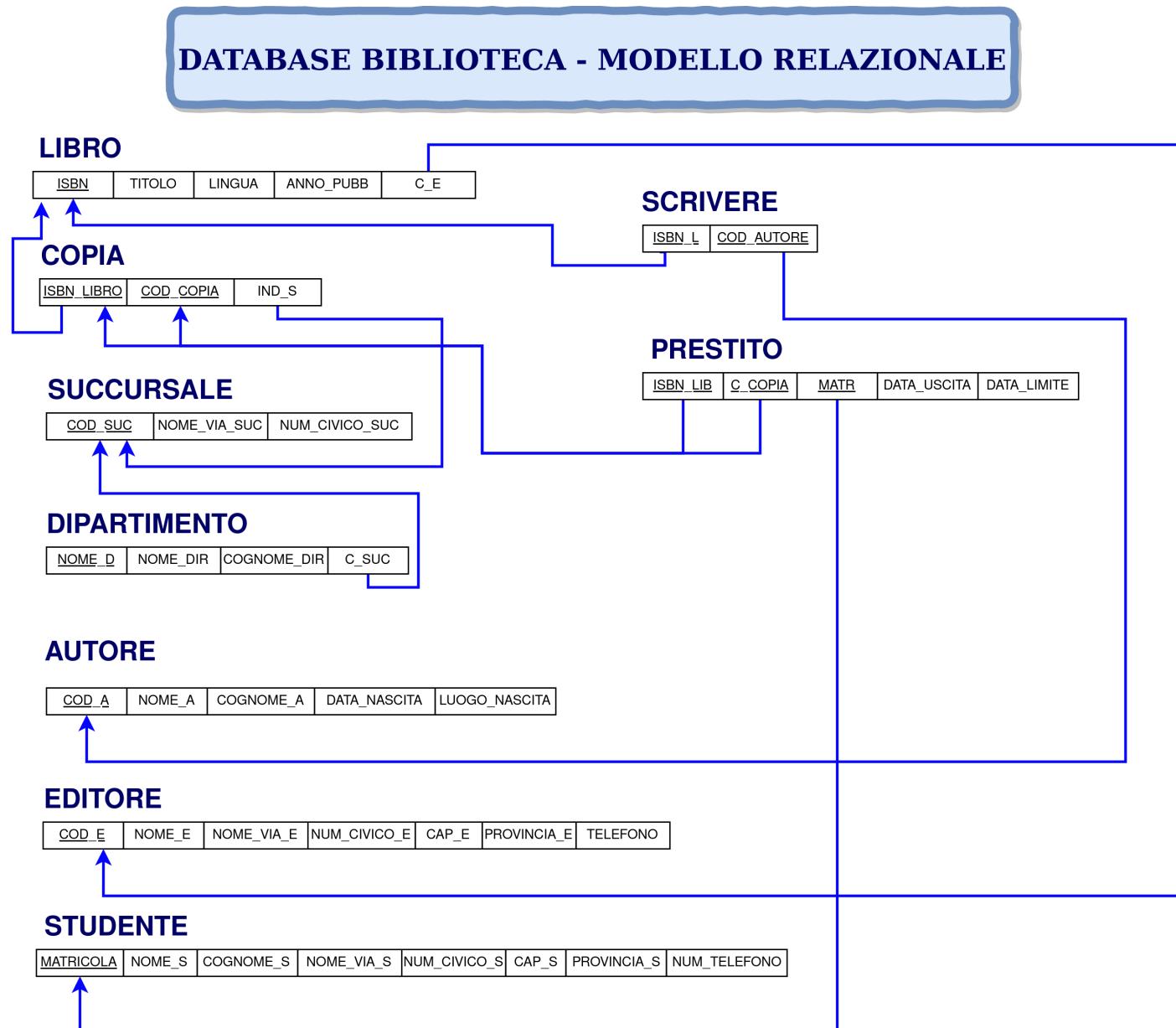


Figure 2: Modello Relazionale

5.1 Normalizzazione

Ci siamo accorti, durante il progetto di uno schema concettuale e la sua conseguente trasformazione in schema relazionale, che vi sono molti potenziali rischi che possono farci prendere la strada sbagliata. Ci vorrebbe, come nel caso del controllo sull'operazione di moltiplicazione (la famosa prova del nove) un qualcosa di simile che ci permetesse di valutare la *bontà* di uno schema prima di procedere. Questo processo esiste, è abbastanza meccanico ed è chiamato **normalizzazione**. In realtà, già in partenza ci siamo abituati a procedere nella stessa direzione in cui punta questo procedimento, per cui abbiamo dovuto fare solo una verifica rapida allo schema. La normalizzazione prevede il raggiungimento di tre stadi successivi, ciascuno dei quali porta lo schema relazionale² ad un livello di ottimizzazione e di eliminazione progressiva dei rischi di *anomalie* (di aggiornamento, cancellazione ed inserimento) e di *ridondanza*. Gli stadi sono almeno 3, corrispondenti a quelli che vengono unanimamente riconosciute come le *3 forme normali* (abbiamo accennato ad una 4°, 5° e di Boyce-Codd, risultanti tuttavia progressivamente eccessive). Già rispettando la 3NF, si coprono la maggior parte delle situazioni. Abbiamo visto come il concetto di normalizzazione sia subordinato a quelli di *Dipendenza Funzionale*, *Dipendenza Transitiva* ecc., che non trattiamo per mancanza di tempo e spazio. Definiamo e motiviamo ognuna delle tre forme normali.

Una relazione è in **prima forma normale (1NF)** quando rispetta i requisiti fondamentali del modello relazionale (già visti in precedenza) che sono:

- tutte le tuple della tabella contengono lo stesso numero di colonne;
- gli attributi rappresentano informazioni elementari;
- i valori che compaiono in una colonna sono dello stesso tipo, cioè appartengono allo stesso dominio;
- ogni riga è diversa da tutte le altre, cioè non ci possono essere due righe con gli stessi valori nelle colonne;
- l'ordine con il quale le righe compaiono nella tabella è irrilevante.

In particolare gli attributi devono essere informazioni **non ulteriormente scomponibili**, né essere gruppi di **attributi ripetuti**. Come già noto dall'ER, abbiamo preferito rendere atomici tutti gli attributi, eccetto quelli che rappresentano una data (nel formato ‘aaaa/mm/gg’). Come detto, essa non è atomica e potremmo separarla in 3 valori, tuttavia è talmente di uso comune che

²Per una miglior visualizzazione dello schema con il software Draw.IO si visiti la relativa [Pagina](#).

si è pensato di rendere disponibile un tipo di dato specifico (**Date**), offerto dal Modello Fisico, con tutti i controlli e gli automatismi necessari. Invece, abbiamo tenuto separate le informazioni rappresentanti un indirizzo. Già dall'ER vediamo infatti che le entità EDITORE, AUTORE e STUDENTE presentano attributi del tipo: **Nome_Via_***, **Num_Civico_***, **CAP_***, **Provincia_***. Se le avessimo tenute tutte assieme, avremmo avuto una grossa stringa e, dal punto di vista semantico, se volessimo fare delle selezioni o dei controlli basati solo su una parte, (Es. sul Num_civico o sul CAP), tutto diventerebbe più problematico. Analogamente, in DIPARTIMENTO, per gestire il Direttore abbiamo predisposto i due attributi **Nome_Dir**, **Cognome_Dir**. Proseguiamo con il processo di normalizzazione.

Una relazione è in **seconda forma normale (2NF)** quando è in 1NF e tutti i suoi attributi non-chiave dipendono dall'intera chiave, cioè non possiede attributi che dipendono soltanto da una parte della chiave. La seconda forma normale elimina la dipendenza parziale degli attributi dalla chiave e riguarda il caso di relazioni con chiavi composte, cioè formate da più attributi.

Le relazioni interessate, ossia che presentano una chiave multicampo sono: COPIA, SCRIVERE, PRESTITO. Dalla nostra analisi, è emerso che ogni attributo appartenente a questa chiave composta determini in modo funzionale i rimanenti attributi non-chiave. Concludendo, abbiamo verificato che lo schema relazionale rispettasse la **terza forma normale (3NF)**, assicurandoci che tutti gli attributi non-chiave di ogni relazione dipendessero direttamente dalle chiave; in altre parole, che nessuno schema di relazione possedesse attributi non-chiave che dipendessero da altri attributi non-chiave.

6 Modello Fisico

A questo punto i dati delle tabelle del precedente modello relazionale sono pronti per essere ubicati negli archivi fisici della memoria secondaria. A questo scopo, abbiamo utilizzato il DBMS **MySQL Workbench**, che ci ha permesso di sfruttare gran parte delle operazioni di: *DDL*, per definire la struttura delle relazioni del database e controllarne gli accessi; *DML*, per modificare i dati contenuti nella base di dati con l'operazione di inserimento. (Le rimanenti operazioni di variazione e cancellazione, assieme alle interrogazioni al database - funzioni di *Query Language* - sono oggetto della prossima sezione). Per la loro realizzazione abbiamo utilizzato il **linguaggio SQL**. Dunque, il primo passo è stato quello di creare il DB e le tabelle con i rispettivi attributi e, per ognuno di

essi specificare il nome, tipo di dato e una tra le varie clausole con le quali è possibile definire: la chiave primaria, le chiavi esterne, l'obbligatorietà e il valore di default di un campo. Frequenti erano frammenti di istruzioni come: ‘CREATE TABLE...’, ‘PRIMARY KEY’, ‘NOT NULL’, ‘DEFAULT’, ‘REFERENCES’, ‘UNIQUE’, ‘ON DELETE’, ‘ON UPDATE’, ‘CASCADE’, ‘SET NULL’ ecc. Considerato che la struttura di una tabella può essere modificata in un secondo momento, ad esempio per inserire una nuova colonna a quelle già esistenti, si è usata l’istruzione ‘ALTER TABLE...’. Per alcune tabelle abbiamo impostato anche le *Referential Integrity Options* viste a lezione. Abbiamo attinto le informazioni da inserire all’interno delle tabelle dai file CSV già preparati concettualmente ed organizzati logicamente, importati in MySQL Workbench tramite il comando ‘LOAD DATA INFILE...’. Infine, per inserire i valori degli attributi nelle righe della tabella abbiamo usato lo statement ‘INSERT INTO...’.

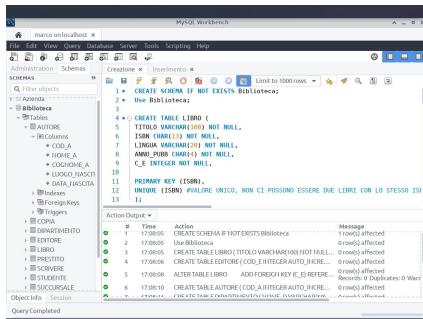


Figure 3: Creazione DB

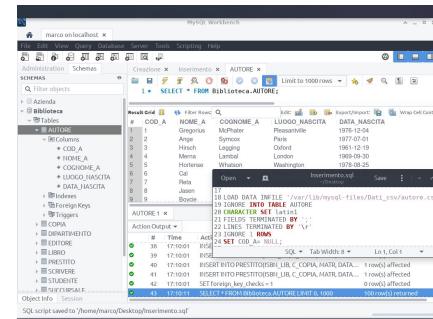


Figure 4: Popolazione DB

7 Algebra Relazionale - Linguaggio SQL

7.1 Statistiche di Base

Di seguito illustriamo le interrogazioni richieste sia in algebra relazionale che in SQL.

- 1]** Elenco di tutti i libri dati in prestito ad un determinato studente in base al suo nome e cognome (Inserendo tali parametri anche parzialmente).

LP \leftarrow LIBRO $\bowtie_{(ISBN_LIB = ISBN)} PRESTITO$

PARZ \leftarrow *LP* $\bowtie_{(MATRICOLA = MATR)}$ *STUDENTE*

$S \leftarrow \sigma_{(NOME_S = '\$NOME_S' \wedge COGNOME_S = '\$COGNOME_S')}(PARZ)$

$TOT \leftarrow \Pi_{(TITOLO, ISBN_LIB, NOME_S, COGNOME_S, DATA_USCITA, DATA_LIMITE)}(S)$

```

1  SELECT TITOLO , ISBN_LIB , NOME_S , COGNOME_S , DATA_USCITA , DATA_LIMITE
2  FROM PRESTITO , LIBRO , STUDENTE
3  WHERE ISBN_LIB = ISBN AND NOME_S LIKE '%$NOME_S%' AND COGNOME_S
4  LIKE '%$COGNOME_S%' AND MATRICOLA= MATR
5

```

- 2] Elenco di tutti gli studenti che hanno in prestito un determinato libro in base al titolo (inserendolo anche solo parzialmente).

$$\begin{aligned}
 LP &\leftarrow \text{LIBRO} \bowtie_{(ISBN_LIB = ISBN)} \text{PRESTITO} \\
 PARZ &\leftarrow LP \bowtie_{(MATRICOLA = MATR)} \text{STUDENTE} \\
 S &\leftarrow \sigma_{(TITOLO = '$TITOLO')}(PARZ) \\
 TOT &\leftarrow \Pi_{(TITOLO, NOME_S, COGNOME_S, DATA_USCITA, DATA_LIMITE)}(S)
 \end{aligned}$$

```

1  SELECT TITOLO , NOME_S , COGNOME_S , DATA_USCITA , DATA_LIMITE
2  FROM PRESTITO , LIBRO , STUDENTE
3  WHERE ISBN_LIB = ISBN AND TITOLO LIKE '%$TITOLO%' AND MATR=MATRICOLA ;
4

```

- 3] Selezione delle 5 lingue più comuni in cui sono scritti i libri presenti nella basi di dati.

$$\begin{aligned}
 TEMP &\leftarrow \rho_{LIBRO(LINGUA, QUANTITA')}(LINGUA \mathcal{F}COUNT(LINGUA)(LIBRO)) \\
 TEMP1 &\leftarrow TEMP \\
 PARZ1 &\leftarrow \rho_{TEMP(LINGUA, QUANTITA')}(LINGUA \mathcal{F}MAX(QUANTITA')(TEMP)) \\
 TEMP2 &\leftarrow TEMP1 - PARZ1 \\
 TEMP3 &\leftarrow TEMP2 \\
 PARZ2 &\leftarrow \rho_{TEMP2(LINGUA, QUANTITA')}(LINGUA \mathcal{F}MAX(QUANTITA')(TEMP2)) \\
 TEMP4 &\leftarrow TEMP3 - PARZ2 \\
 TEMP5 &\leftarrow TEMP4 \\
 PARZ3 &\leftarrow \rho_{TEMP4(LINGUA, QUANTITA')}(LINGUA \mathcal{F}MAX(QUANTITA')(TEMP4)) \\
 TEMP6 &\leftarrow TEMP5 - PARZ3 \\
 TEMP7 &\leftarrow TEMP6 \\
 PARZ4 &\leftarrow \rho_{TEMP6(LINGUA, QUANTITA')}(LINGUA \mathcal{F}MAX(QUANTITA')(TEMP6)) \\
 TEMP8 &\leftarrow TEMP7 - PARZ4 \\
 PARZ5 &\leftarrow \rho_{TEMP8(LINGUA, QUANTITA')}(LINGUA \mathcal{F}MAX(QUANTITA')(TEMP8)) \\
 RIS &\leftarrow PARZ1 \cup PARZ2 \cup PARZ3 \cup PARZ4 \cup PARZ5
 \end{aligned}$$

```

1  SELECT LINGUA , COUNT(LINGUA) AS QUANTITA

```

```

2   FROM LIBRO
3   GROUP BY LINGUA
4   ORDER BY QUANTITA DESC LIMIT 5;
5

```

4] Selezione dell'/degli autore/i che ha/hanno scritto più libri.

$$\begin{aligned}
 A &\leftarrow \text{AUTORE} \bowtie_{(COD_A = COD_AUTORE)} \text{SCRIVERE} \\
 PARZ &\leftarrow_{(NOME_A, COGNOME_A)} \mathcal{F}\text{COUNT}_{(COD_A)}(A) \\
 PARZ1 &\leftarrow \rho_{(NOME, COGNOME, QUANTITA')}(PARZ) \\
 RIS &\leftarrow \sigma_{(QUANTITA' = \mathcal{F}\text{MAX}(QUANTITA'))}(PARZ1)
 \end{aligned}$$

```

1   SELECT NOME_A, COGNOME_A, COUNT(COD_A) AS QUANTITA
2   FROM AUTORE, SCRIVERE
3   WHERE COD_AUTORE = COD_A
4   GROUP BY NOME_A, COGNOME_A
5   HAVING QUANTITA = (
6       SELECT MAX(A.QUANTITA)
7       FROM (
8           SELECT COUNT(COD_A) AS QUANTITA
9           FROM AUTORE, SCRIVERE
10          WHERE COD_AUTORE = COD_A
11        GROUP BY NOME_A, COGNOME_A) AS A)
12

```

5] Selezione dell'/degli editore/i che ha/hanno pubblicato più libri.

$$\begin{aligned}
 E &\leftarrow \text{EDITORE} \bowtie_{(COD_E = C_E)} \text{LIBRO} \\
 PARZ &\leftarrow_{(NOME_E)} \mathcal{F}\text{COUNT}_{(COD_E)}(E) \\
 PARZ1 &\leftarrow \rho_{(NOME, QUANTITA')}(PARZ) \\
 RIS &\leftarrow \sigma_{(QUANTITA' = \mathcal{F}\text{MAX}(QUANTITA'))}(PARZ1)
 \end{aligned}$$

```

1   SELECT NOME_E, COUNT(COD_E) AS QUANTITA
2   FROM EDITORE, LIBRO
3   WHERE COD_E = C_E
4   GROUP BY NOME_E
5   HAVING QUANTITA = (
6       SELECT MAX(A.QUANTITA)
7       FROM (
8           SELECT COUNT(COD_E) AS QUANTITA

```

```

9   FROM EDITORE , LIBRO
10  WHERE COD_E = C_E
11  GROUP BY NOME_E) AS A)
12

```

7.2 Interrogazioni Aggiuntive

- 1] Ricerca della succursale che possiede il maggior numero di libri.

```

S←COPIA▷(COD_SUC = IND_S)SUCCURSALE
PARZ←NOME_VIA_SUC, NUM_CIVICO_SUC, FCOUNT(IND_S)(S)
PARZ1←ρ(VIA, N_CIVICO, QUANTITA')(PARZ)
RIS←(QUANTITA' = FMAX(QUANTITA')(PARZ1))(PARZ1)

```

```

1   SELECT CONCAT_WS(' ',NOME_VIA_SUC,NUM_CIVICO_SUC) AS SUCCURSALE ,
2     COUNT(IND_S) AS QUANTITA
3   FROM SUCCURSALE,COPIA
4   WHERE COD_SUC=IND_S
5   GROUP BY NOME_VIA_SUC,NUM_CIVICO_SUC
6   HAVING QUANTITA = (
7     SELECT MAX(A.QUANTITA)
8     FROM (
9       SELECT COUNT(IND_S) AS QUANTITA
10      FROM COPIA,SUCCURSALE
11     WHERE IND_S=COD_SUC
12     GROUP BY NOME_VIA_SUC,NUM_CIVICO_SUC ) AS A)
13

```

- 2] Ricerca dell'anno in cui sono stati pubblicati il minor numero di libri e relativa quantità.

```

PARZ←ANNO_PUBB FCOUNT(ANNO_PUBB)(LIBRO)
PARZ1←ρ(ANNO,QUANTITA')(PARZ)
RIS←(QUANTITA' = FMIN(QUANTITA')(PARZ1))(PARZ1)

1   SELECT ANNO_PUBB , COUNT(ANNO_PUBB) AS QUANTITA
2   FROM LIBRO
3   GROUP BY ANNO_PUBB
4   HAVING QUANTITA = (
5     SELECT MIN(A.QUANTITA)

```

```

6   FROM (
7     SELECT COUNT(ANNO_PUBB) AS QUANTITA
8     FROM LIBRO
9     GROUP BY ANNO_PUBB) AS A)
10

```

3] Elenco degli studenti che sono in ritardo con la restituzione di un prestito.

$$R \leftarrow STUDENTE \bowtie_{(MATR = MATRICOLA)} PRESTITO$$

$$PARZ \leftarrow R \bowtie_{(ISBN = ISBN_LIB)} LIBRO$$

$$S \leftarrow \sigma_{(DATA_LIMITE < \text{FCURDATE}())}(PARZ)$$

$$RIS \leftarrow \Pi_{(COGNOME_S, NOME_S, DATA_LIMITE, TITOLO)}(S)$$

```

1   SELECT COGNOME_S, NOME_S, DATA_LIMITE, TITOLO
2   FROM STUDENTE, PRESTITO, LIBRO
3   WHERE MATR=MATRICOLA AND ISBN=ISBN_LIB AND DATA_LIMITE < curdate()
4   ORDER by DATA_LIMITE
5   --data odierna;

```

4] Elenco dei libri scritti in media da un autore. (Attenzione, il risultato andrà opportunamente arrotondato).

$$R \leftarrow COD_AUTORE \mathcal{F}COUNT_{(ISBN_L)}(SCRIVERE)$$

$$CONTEGGIO \leftarrow \rho_{(COD_AUTORE, LIBRI_SCRITTI)}(R)$$

$$PARZ \leftarrow \mathcal{F}AVG(LIBRI_SCRITTI)(CONTEGGIO)$$

$$RIS \leftarrow \rho_{(MEDIA)}(PARZ)$$

```

1   SELECT AVG (CONTEGGIO.LIBRI_SCRITTI ) AS MEDIA
2   FROM ( SELECT COUNT( ISBN_L ) AS LIBRI_SCRITTI
3   FROM SCRIVERE
4   GROUP BY COD_AUTORE ) AS CONTEGGIO
5

```

5] Elenco dei libri scritti in una determinata lingua presenti nella succursale di interesse.

$$R \leftarrow LIBRO \bowtie_{(ISBN = ISBN_LIBRO)} COPIA$$

$$PARZ \leftarrow R \bowtie_{(IND_S = COD_SUC)} SUCCURSALE$$

$$RIS \leftarrow \sigma_{(LINGUA = '$LINGUA', COD_SUC = '$SUCCURSALE')}(\Pi_{(TITOLO, ISBN,})$$

$$ANNO_PUBB, LINGUA, NOME_VIA_SUC, NUM_CIVICO_SUC}(PARZ))$$

```

1  SELECT DISTINCT TITOLO , ISBN , ANNO_PUBB , LINGUA ,
2    CONCAT_WS( ' , ' , NOME_VIA_SUC , NUM_CIVICO_SUC) AS SEDE
3  FROM LIBRO , COPIA , SUCCURSALE
4 WHERE ISBN_LIBRO = ISBN AND IND_S = COD_SUC AND LINGUA = '$LINGUA'
5   AND COD_SUC = '$SUCCURSALE'
6

```

- 6] Elenco degli studenti che hanno preso in prestito i libri di una determinata casa editrice.

$$\begin{aligned}
R &\leftarrow \text{LIBRO} \bowtie_{(ISBN = ISBN_LIB)} \text{PRESTITO} \\
PARZ1 &\leftarrow R \bowtie_{(MATR = MATRICOLA)} \text{STUDENTE} \\
PARZ2 &\leftarrow PARZ1 \bowtie_{(C_E = COD_E)} \text{EDITORE} \\
RIS &\leftarrow \sigma_{(COD_E = '$EDITORE')}(\Pi_{(NOME_S, COGNOME_S, TITOLO, NOME_E)}(PARZ2))
\end{aligned}$$

```

1  SELECT NOME_S , COGNOME_S , TITOLO , NOME_E
2  FROM STUDENTE , LIBRO , PRESTITO , EDITORE
3  WHERE MATR = MATRICOLA AND ISBN = ISBN_LIB
4  AND C_E = COD_E AND COD_E = '$EDITORE'
5

```

- 7] Elenco del numero di libri pubblicati dall'anno scelto fino ad oggi da ciascun editore.

$$\begin{aligned}
TEMP &\leftarrow \text{LIBRO} \bowtie_{(COD_E = C_E)} \text{EDITORE} \\
TEMP1 &\leftarrow \sigma_{(ANNO_PUBB > '$ANNO_PUBB')}(TEMP) \\
RIS &\leftarrow \text{NOME_E} \text{FCOUNT}_{(ISBN)}(TEMP1)
\end{aligned}$$

```

1  SELECT NOME_E , COUNT( ISBN )
2  FROM LIBRO , EDITORE
3  WHERE COD_E = C_E AND ANNO_PUBB > '$ANNO_PUBB'
4  GROUP BY NOME_E
5  ORDER BY NOME_E ASC
6

```

- 8] Elenco dei libri presenti in una data sede e relativa quantità con cui sono presenti (Inserendone anche solo parzialmente il titolo).

$R \leftarrow COPIA \bowtie_{(ISBN_LIBRO = ISBN)} LIBRO$

$PARZ \leftarrow R \bowtie_{(IND_S = COD_SUC)} SUCCUSALE$

$PARZ1 \leftarrow \sigma_{(TITOLO = '$TITOLO')}(PARZ)$

$PARZ2 \leftarrow TITOLO, ANNO_PUBB, NOME_VIA_SUC, NUM_CIVICO_SUC,$

$LINGUA \mathcal{F}COUNT_{(ISBN_LIBRO)}(PARZ1)$

$RIS \leftarrow \rho_{(TITOLO, ANNO_PUBB, VIA, N_CIVICO, LINGUA, QUANTITA')}(PARZ2)$

```

1  SELECT TITOLO , ANNO_PUBB , CONCAT_WS( ' , ' , NOME_VIA_SUC , NUM_CIVICO_SUC) AS SEDE ,
2    LINGUA , COUNT(ISBN_LIBRO) AS QUANTITA
3  FROM COPIA , LIBRO , SUCCURSALE
4 WHERE ISBN_LIBRO = ISBN AND IND_S=COD_SUC AND TITOLO LIKE '%$TITOLO%'
5 GROUP BY TITOLO , ANNO_PUBB , SEDE , LINGUA
6 ORDER BY TITOLO ;
7

```

9] Elenco dei dipartimenti che condividono la stessa succursale.

$R \leftarrow DIPARTIMENTO \bowtie_{(C_SUC = COD_SUC)} SUCCURSALE$

$PARZ \leftarrow \Pi_{(NOME_D, NOME_VIA_SUC, NUM_CIVICO_SUC)}(R)$

$RIS \leftarrow \rho_{(DIPARTIMENTO, VIA, N_CIVICO)}(R)$

```

1  SELECT GROUP_CONCAT( NOME_D SEPARATOR ' , ') AS DIPARTIMENTI ,
2    CONCAT_WS( ' , ' , NOME_VIA_SUC , NUM_CIVICO_SUC) AS SEDE
3  FROM DIPARTIMENTO , SUCCURSALE
4 WHERE C_SUC=COD_SUC
5 GROUP BY SEDE ;
6

```

8 Pagine Web



Figure 5: Schema del Progetto Web

8.1 Descrizione delle Pagine Web

L'accesso al database avviene grazie alla pagina ‘conessione.php’ la quale, tramite il proprio username e password, tenta l'effettiva connessione. Nel caso non vada a buon fine è riportato un messaggio di errore con associato tipo di errore. L'applicativo web realizzato in PHP è composto da un **barra di navigazione** contenente un menu che permette l'accessibilità alle varie pagine del sito. A piè di pagina, si trovano dei collegamenti ai social della biblioteca e all'email per fornire eventuali informazioni, suggerimenti o reclami. La navbar e il footer sono presenti (richiamati) in ogni pagina e sono definiti tramite la propria funzione in ‘function.php’.

HOME

Nella homepage (‘index.php’) è presente una sezione chiamata “CATALOGO” grazie alla quale è possibile effettuare la ricerca di un determinato libro. Una volta inserito il titolo e scelto la propria sede bibliotecaria (se non selezionata saranno visualizzate tutte le sedi) si sarà reindirizzati in una pagina (‘ricerca_catalogo.php’) in cui sarà visualizzato il risultato della ricerca del libro con affiancata la *quantità* con cui è presente. Per quest'ultima era espressamente richiesto di tenerne traccia, perciò abbiamo predisposto il conteggio delle singole copie presenti in ciascuna sede osservando che, grazie alla presenza della chiave esterna IND_S (riferita alla chiave primaria COD_SUC della tabella SUCCUSALE) nella tabella COPIA, possiamo risalire a dove è contenuto ciascun libro. Nella parte sottostante la sezione “CATALOGO” troviamo una tabella che elenca i vari Dipartimenti che condividono la stessa succursale della biblioteca.

GESTIONE STUDENTI

Nel menu, a destra della “HOME” troviamo “GESTIONE STUDENTI”, la cui implementazione risiede in ‘gestione_studenti.php’. Essa offre funzionalità come:

- l'*inserimento* di un nuovo studente all'interno del database: immettendo la propria matricola fornita dall'università e i propri dati personali. Una volta confermato l'inserimento si sarà indirizzati alla pagina ‘insert_studente.php’ in cui si sarà avvisati del corretto inserimento o meno. In caso affermativo si visualizzeranno i dati inseriti nel sistema.
- l'*aggiornamento* di uno studente: selezionando la propria matricola si potranno aggiornare i propri dati personali (indirizzo e numero civico, CAP, provincia, numero di telefono). Una volta inseriti i dati desiderati, si sarà rimandati alla pagina ‘update_studente.php’ nella quale si visioneranno le modifiche apportate.

- la *cancellazione* di uno studente: scegliendo la propria matricola dal menu di selezione sarà possibile cancellare il proprio contatto dal sistema (si potrà solo se non ci sono dei prestiti in corso, in tal caso comparirà un messaggio di errore, inoltre sarà riportato l'elenco dei prestiti attualmente in corso). Una volta confermata la matricola con l'apposito pulsante si sarà reindirizzati in ‘`delete_studente.php`’ dove si sarà informati della corretta cancellazione o, in caso contrario, si visualizzerà la lista dei prestiti in corso.

■ GESTIONE PRESTITI

Accanto a “GESTIONE STUDENTI” scorgiamo “GESTIONE PRESTITI” (‘`gestione_prestiti.php`’), che ci permette:

- l’*inserimento* di un nuovo prestito: immettendo il titolo del libro desiderato e scegliendo una copia del libro dal menu di selezione (in cui sono presenti solo le copie disponibili, ossia quelle non attualmente in prestito). Una volta confermato l’inserimento si sarà reindirizzati alla pagina ‘`insert_prestito.php`’ nella quale si sarà avvisati sul corretto inserimento o meno del prestito e saranno elencati: le informazioni sul libro, il codice del prestito, la data di uscita e la data limite standard di 30 giorni.
- la proroga di un prestito (*modifica*): scegliendo la propria matricola dal menu di selezione (scelta tra gli studenti presenti nel sistema), selezionando il titolo del libro (nel menu di selezione saranno visibili solo i titoli dei libri presi in prestito da quel determinato studente) e il relativo codice della copia e inserendo il tempo di proroga (sarà possibile scegliere tra 10,15 o 30 giorni). Dopo aver confermato, in una nuova pagina (‘`update_prestito.php`’) sarà mostrata la nuova data limite per la restituzione e i riferimenti del prestito in oggetto.
- restituzione di un prestito(*cancellazione*): scegliendo la propria matricola, il titolo del libro e il relativo codice copia (sono presenti solo i titoli dei libri e relativo codice presi in prestito da quel determinato studente) del libro che si vuole restituire. Una volta confermata la volontà di effettuare tale restituzione si sarà rediretti in una pagina (‘`delete_prestito.php`’) che permetterà di visualizzare un messaggio di errore, nel caso di scorrettezze nell’inserimento dei dati, oppure di vedere un messaggio di corretta restituzione accompagnato dalle informazioni concernenti il prestito che si sta considerando.

La gestione dei prestiti è stata in parte corredata grazie all’ausilio di Javascript, in particolare grazie al suo framework JQuery il quale, sfruttando la tecnologia AJAX (in ‘`load_data.php`’)

ha permesso l'aggiornamento della pagina web senza dover effettuare un nuovo caricamento della stessa. Nello specifico è stata utilizzata in modo tale da far variare il contenuto dei menu di selezione presenti nei form (all'interno di ‘gestione_prestiti.php’) in modo dinamico, ossia in base alla scelta della propria matricola o del titolo del libro; di conseguenza, in ‘load_data.php’ sono presenti le interrogazioni al database che saranno eseguite relativamente alla selezione effettuata dall'utente.

Tutto ciò è stato fatto principalmente per evitare problematiche, come la restituzione di un prestito che in realtà non appartiene ad un determinato studente, oppure richiedere una copia di un libro che è già in prestito ad un altro studente.

RICERCA

Proseguendo, si trova l'opzione “RICERCA” (‘ricerca.php’) contenente parte delle statistiche di base espressamente richieste dalla biblioteca universitaria. Essa consente di effettuare:

- la ricerca dei libri in prestito ad un determinato studente: inserendo anche solo parzialmente il nome e cognome di uno studente si sarà reindirizzati alla pagina ‘ricerca_studente.php’ in cui sarà possibile consultare la situazione dei prestiti (titolo del libro, ISBN, nome e cognome dello studente, data uscita e data limite) cioè una tabella contenente il risultato prodotto dalla ricerca o, eventualmente un messaggio nel caso in cui la ricerca non abbia prodotto risultati.
- la ricerca degli studenti che hanno in prestito un determinato libro: inserendo anche solo parzialmente il titolo di un libro ci si sposterà nella pagina ‘ricerca_libro.php’. Nel caso la ricerca vada a buon fine, si visualizzerà una tabella contenente gli studenti che hanno preso in prestito quel determinato libro risultante dalla ricerca, il titolo, la data di uscita e la data limite entro la quale restituirlo.

STATISTICHE

Se spostiamo il focus sul quinto bottone del menu di navigazione, possiamo notare la comparsa di una sorta di menu a tendina, nel quale è possibile consultare le “STATISTICHE” in due distinte modalità di visualizzazione “TABELLE” (‘tabelle.php’) o “GRAFICI” (‘grafici.php’). Le due pagine si differenziano solo per il formato di visualizzazione, ma entrambe riportano le stesse statistiche di base:

- Le 5 lingue più comuni in cui sono scritti i libri presenti nella base di dati.

- L'autore che ha scritto più libri.
- L'editore che ha pubblicato più libri.

In ‘tabelle.php’ sono presenti inoltre gli output di alcune statistiche aggiuntive e i form delle rimanenti:

- Succursale che possiede il maggior numero di libri.
- Anno in cui sono stati pubblicati il minor numero di libri.
- Elenco degli studenti che sono in ritardo con la restituzione del prestito
- Media dei libri scritti da ciascun autore
- Elenco dei libri scritti in una determinata lingua presenti nella succursale di interesse (‘libri_lingua_succ.php’).
- Elenco degli studenti che hanno preso in prestito i libri di una determinata casa editrice (‘studenti_editore.php’).
- Elenco del numero di libri pubblicati da ciascun editore dall'anno di pubblicazione scelto fino ad oggi (‘libro_editore.php’).

In `index.php` troviamo anche l' elenco dei dipartimenti che condividono la stessa succursale. Infine in ‘ricerca_catalogo.php’ abbiamo la query che tiene traccia delle quantità di tutti libri presenti in una data sede, in base al titolo parziale o totale. Cliccando sull’ “ancora” alla pagina ‘grafici.php’ denominata “GRAFICI”, osserviamo una serie di diagrammi che sono stati realizzati grazie a Chart.js, una libreria di JavaScript open source che permette di ottenere grafici dinamici che si aggiornano in base alle informazioni contenute all'interno della nostra base di dati, rendendo più intuitiva la loro interpretazione. Ogni grafico possiede la relativa pagina: ‘lingua.php’, ‘autore.php’, ‘editori.php’ in cui è effettuata la connessione al database, è poi specificata la query da eseguire e salvato il risultato che si vuole mostrare sul grafico.

9 Conclusioni

Siamo giunti al capolinea di questo progetto.

Riportiamo alcuni accorgimenti: in talune interrogazioni in algebra relazionale non abbiamo usato l'operatore di proiezione, in quanto come visto a lezione, tale operazione è implicita quando si utilizza l'operatore di funzione aggregata.

Alcune fonti che ci hanno permesso di realizzare tutto ciò sono:

- Overleaf
- W3School
- Sito del PHP
- Sito di MySQL
- Slide del corso.

Grazie per l'attenzione,

Enrico Manfrin, Marco Beltrame