

# BIBLIOTECA UNIFE

Relazione riguardante il progetto di basi di dati per l'anno 2019/2020. Tutto il materiale è disponibile all'indirizzo: <https://github.com/Giulio987/ProgettoDatabase.git>

Il progetto è stato unicamente realizzato da:

**Milani Giulio**                      **Matricola: 144195**

**Davide Polidori**                      **Matricola: 149473**

Software utilizzato per la collaborazione in real time: "ATOM" con il plugin "teletype". I vari aggiornamenti sono stati pubblicati su Github periodicamente quando entrambi i componenti del gruppo si riunivano tramite il software "Skype".

Il dump del database è disponibile all'interno della cartella DUMP.

## INTRODUZIONE:

L'Università degli Studi di Ferrara ha commissionato la costruzione di uno strumento informatico per la gestione della biblioteca universitaria frammentata in più dipartimenti.

E' stato creato un applicativo rivolto al personale che permetta di gestire in modo facile e veloce la situazione dei prestiti e degli studenti all'interno delle varie strutture e che dia la possibilità di consultare informazioni utili al fine della corretta gestione dell'apparato universitario.

Entrando nello specifico questo strumento è costituito dal database che contiene tutte le informazioni della biblioteca, esso è poi collegato all'applicativo web creato tramite i linguaggi: HTML, CSS, JAVASCRIPT e PHP.

Le funzionalità dell'applicativo web sono suddivise in 3 macro categorie:

1. La prima, dedicata alla gestione delle informazioni riguardanti gli studenti, che a sua volta è suddivisa in 3 parti:
  - Registrazione di un nuovo studente: inserimento dei dati dello studente che viene quindi registrato nel database. Questa procedura avviene nel momento in cui lo studente entra a far parte dell'università e non la prima volta in cui esso utilizza il sistema bibliotecario.
  - Modifica delle informazioni relative allo studente: una volta ricercato lo studente tramite il numero di matricola, vengono visualizzati i suoi dati che possono essere quindi modificati.
  - Rimozione dello studente dal database: sempre attraverso la ricerca tramite matricola, le sue informazioni vengono visualizzate in forma tabellare per poi procedere con l'eventuale eliminazione. Questa operazione avviene nel momento in cui esso non farà più parte dell'università (ad esempio per un trasferimento ad un'altra università).
2. La seconda, dedicata alla gestione dei prestiti, suddivisa in 4 parti:
  - Concessione del prestito a uno studente: qui si inseriscono i dati relativi allo studente beneficiario del prestito e al libro che la biblioteca vuole concedere. E' inoltre presente la data odierna che sarà quella dell'inizio del prestito.
  - Modifica delle informazioni associate al prestito: inserendo i dati relativi ad un prestito in corso e si ottengono la data di inizio e di fine di esso. E' stato ritenuto opportuno

che l'unica informazione modificabile in questa sezione sia la data di rientro poiché le altre informazioni (tranne la data di inizio prestito) sono collegate ad altre entità vincolate.

- Restituzione del libro alla biblioteca: si inseriscono i dati relativi ad un prestito in corso per effettuare la restituzione presso la biblioteca. Qui vengono eliminate le informazioni riguardanti il prestito associato allo studente nella base di dati.
- Situazione dei prestiti: viene visualizzato un elenco in cui sono presenti TUTTI i prestiti all'interno del database come richiesto alla consegna.

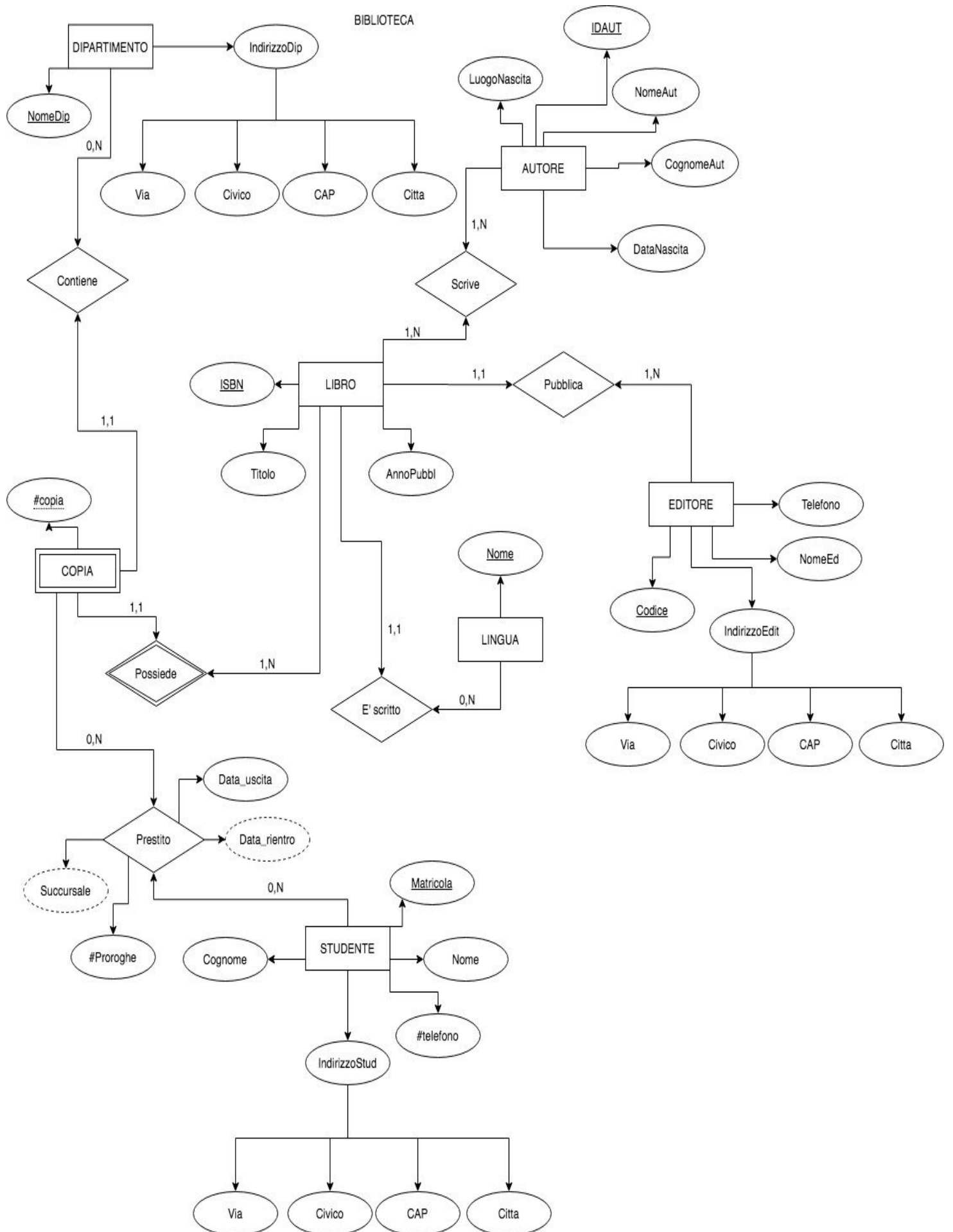
3. La terza, dedicata agli strumenti per ottenere informazioni utili dal database, essa contiene:

- L'elenco del/i libro/i dati in prestito a uno studente: inserendo anche solo parzialmente il nome e di uno studente vengono visualizzati i suoi dati e i suoi prestiti.
- L'elenco dello/gli studente/i a cui è stato dato in prestito un determinato libro: inserendo anche solo parzialmente il nome di un libro vengono visualizzati tutti gli studenti che hanno in prestito una sua copia.
- L'elenco di alcune statistiche di base: sono visualizzate le 5 lingue più frequenti in cui sono scritti i libri, l'autore con più libri scritti (se ci sono più autori con gli stessi libri scritti vengono visualizzati tutti) e l'editore che ha pubblicato più libri (se ci sono più editori con lo stesso numero di libri pubblicati vengono visualizzati tutti).
- Alcuni strumenti aggiuntivi per ricavare informazioni utili: in questa sezione si ha la possibilità di inserire il nome di un libro per ricavarne le copie presenti e in quale dipartimento esse si trovano. Vi sono poi 4 tabelle, contenenti i primi 15 studenti con più prestiti attivi, i dipartimenti con più libri, i prestiti in scadenza e i prestiti scaduti. Infine è possibile inserire una data per ricavare le informazioni su tutti prestiti effettuati quel giorno.

## **SCHEMA CONCETTUALE:**

Verrà ora presentato lo Schema E/R seguito dalla conseguente traduzione in schema Relazionale:

# BIBLIOTECA



E' stato considerato opportuno per la creazione della biblioteca universitaria la creazione delle entità:

- DIPARTIMENTO: ovvero l'entità associata alle informazioni sui vari dipartimenti relativi all'università di Ferrara, ognuno con una propria sede e biblioteca al suo interno.
- STUDENTE: è l'entità riguardante le informazioni degli studenti ovvero il nome, il cognome, il numero di telefono, l'indirizzo (attributo composto suddiviso in via, civico, CAP e città) e la matricola che funge da chiave primaria poiché distingue univocamente ogni studente.
- AUTORE: è l'entità riguardante le informazioni dei vari autori che hanno scritto almeno un libro di quelli presenti nel database, avente come attributi il nome, il cognome, il luogo di nascita, la data di nascita e l'identificativo unico riguardante l'autore di cui si riporteranno spiegazioni in seguito
- LIBRO: è l'entità riguardante le informazioni dei libri ovvero il titolo, l'anno di pubblicazione e l'ISBN che funge da chiave primaria poiché distingue univocamente ogni libro.
- EDITORE: è l'entità riguardante le informazioni degli editori ovvero il nome dell'editore, il suo telefono, l'indirizzo (attributo composto suddiviso in via, civico, CAP e città) e il codice dell'editore che funge da chiave primaria poiché distingue univocamente ogni editore.
- COPIA: l'entità debole copia è univocamente identificata tramite l'ISBN del libro di cui essa è copia e dal numero della copia (chiave parziale) associato, per come è stata pensata non necessita di altre informazioni.
- LINGUA: è l'entità riguardante le informazioni sulle lingue ed è formata solo dal nome della lingua che funge anche da chiave primaria.

Volgendo lo sguardo alle relazioni tra le entità, DIPARTIMENTO è collegato a COPIA tramite una relazione con cardinalità 1 a N, una copia infatti può essere contenuta solamente in un dipartimento mentre un dipartimento può contenere più copie dei libri. COPIA è a sua volta collegata a STUDENTE tramite la relazione "PRESTITO" con cardinalità N a N poiché una copia può essere prestata da 0 a N studenti (in momenti diversi) e uno studente può avere in prestito da 0 a N copie. La relazione PRESTITO è caratterizzata dalla data di uscita (data di inizio del prestito), dalla data di rientro (attributo derivato dalla data di uscita + 30 giorni e dal numero di proroghe), dalla succursale (attributo derivato poiché in relazione con DIPARTIMENTO il suo nome sarà chiave esterna in copia) e dal numero di proroghe (quante volte uno studente ha rinnovato il prestito) che può essere massimo 2 (controllo eseguito su php), ogni proroga è di 15 giorni.

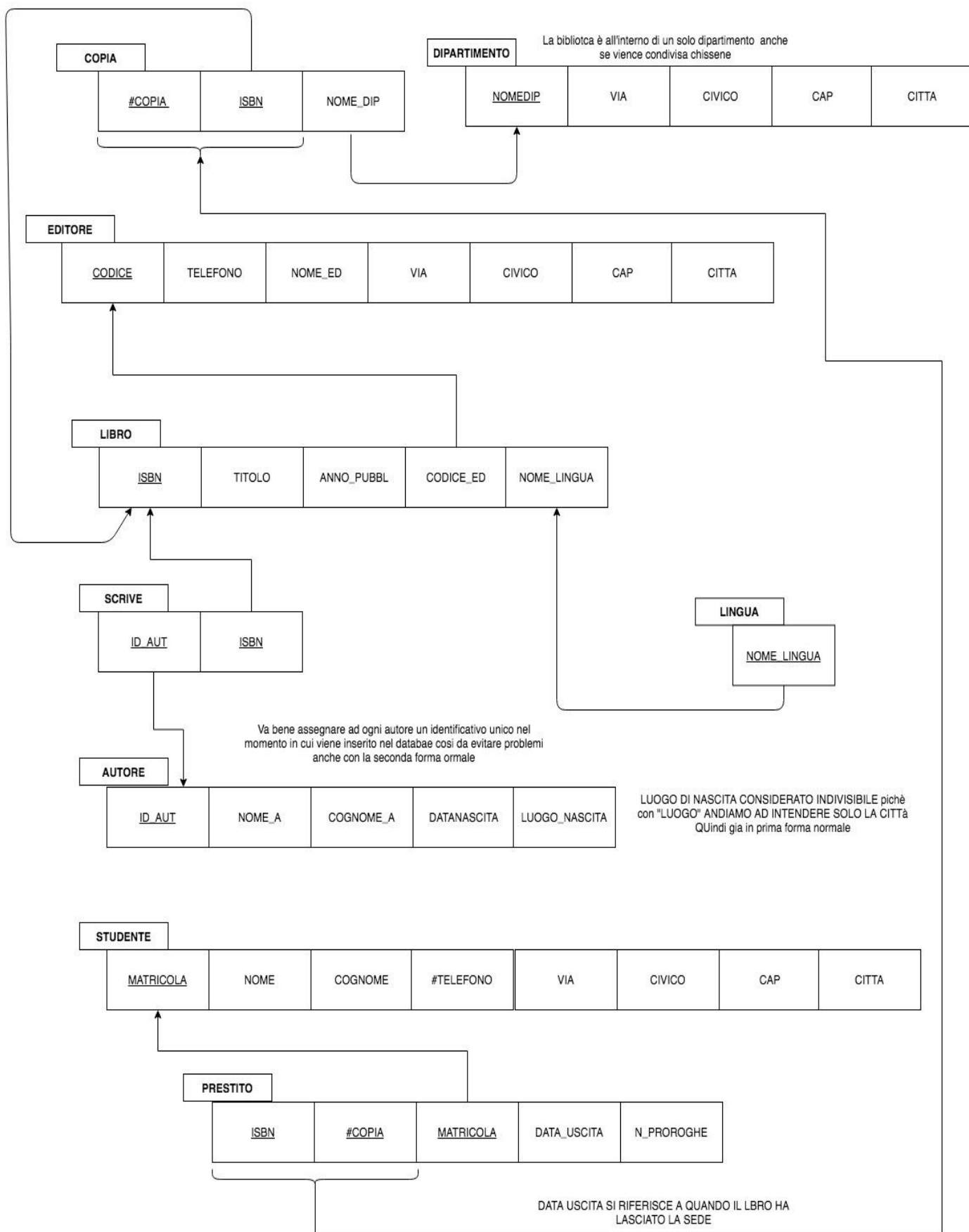
La scelta di inserire PRESTITO come relazione tra 2 entità e non come entità debole è avvenuta poiché se fosse stato fatto la relazione identificante sarebbe stata con COPIA per ottenere ISBN E N\_COPIA come chiavi primarie nonché esterne. A sua volta però COPIA è già un'entità debole quindi il problema era se mettere un'Entità debole in relazione identificante con un'altra entità debole. E' quindi stato deciso di lasciare PRESTITO come relazione N a N avendo poi cura di impedire la concessione di libri già in prestito ad altri studenti tramite appositi controlli PHP. Avendo così ISBN, MATRICOLA e NUMERO\_COPIA come chiavi primarie viene automaticamente impedito l'inserimento di un prestito che coinvolga lo stesso studente con la stessa copia del libro. Inoltre è stato deciso di non inserirla come entità forte poiché, in mancanza di un vero e proprio identificativo, un progressivo unico per ogni prestito nuovo sarebbe sembrata una forzatura evitabile, avendo di continuo nuovi prestiti, inoltre non è sembrato essenziale mantenere i prestiti passati.

COPIA è collegata anche a LIBRO tramite la relazione "possiede" con cardinalità 1 a N, un libro infatti può avere da 1 a N copie mentre una copia può essere solamente di un 1 libro. LIBRO è a

sua volta collegata ad AUTORE con una relazione “scrive” di cardinalità N a N, un libro infatti può essere scritto da 1 a più autori (vedi ad esempio i libri scolastici) ed un autore può scrivere da 1 ad N libri. LIBRO è collegato anche ad EDITORE tramite la relazione “pubblica” di cardinalità 1 a N poiché un libro può essere pubblicato da 1 solo editore mentre un editore può pubblicare da 1 a N libri. Infine LIBRO è collegato a LINGUA tramite la relazione “E’ SCRITTO” di cardinalità 1 a N, infatti un libro può essere scritto in una sola lingua (poiché se si cambia lingua cambia conseguentemente anche l’ISBN) e una lingua è presente in 0 o più libri.

## **SCHEMA RELAZIONALE:**

Passando dal diagramma ER allo schema relazionale tutte le entità sono diventate relazioni che a loro volta avranno una chiave primaria ed eventuali chiavi esterne. Le relazioni 1 a N ottengono come chiavi esterne dalla parte di N le chiavi primarie dell’altra, come nello specifico caso di DIPARTIMENTO e COPIA in cui quest’ultima ottiene come chiave esterna il nome del Dipartimento. Anche tra EDITORE e LIBRO e tra LINGUA e LIBRO, LIBRO ottiene come chiavi esterne il codice dell’editore e la lingua in cui esso è stato scritto. Tra LIBRO e COPIA, in cui invece quest’ultima ottiene l’ISBN che diventa non solo chiave esterna ma anche chiave primaria insieme a N\_COPIA(chiave parziale) essendo COPIA un’entità debole. Le relazioni N a N diventano una nuova relazione che avrà come chiave primaria la chiave composta dalle chiavi primarie di entrambe le relazioni. Questo accade tra LIBRO e AUTORE con la tabella SCRIVE formata da ID\_AUT ed ISBN che sono chiavi esterne e a loro volta formano la chiave primaria di Scrive. Con COPIA e STUDENTE accade la stessa cosa in PRESTITO, la quale ha però anche DATA\_USCITA e N\_PROROGHE come attributi.



Parlando ora di forme normali:

Durante la progettazione si è cercato di rispettare tutte e 3 le forme normali: una scelta di progettazione, data la limitatezza di autori, è stata quella di assegnare a tale Entità/Relazione un progressivo ID\_AUT generato arbitrariamente, conseguentemente sono anche stati evitati i problemi relativi alle dipendenze funzionali

- La prima forma normale è stata rispettata poiché sono stati inseriti unicamente attributi atomici e quindi indivisibili: ad esempio l'indirizzo è stato spezzato in tutte le parti che lo compongono; per LUOGO\_NASCITA in AUTORE si intende solamente la città in cui esso è nato per cui non è stato necessario dividerlo ulteriormente.
- La seconda forma normale è stata rispettata poiché non ci sono relazioni in cui gli attributi non chiave dipendono funzionalmente da una parte della chiave e non dalla chiave completa: ad esempio in prestito il numero di proroghe e la data di inizio prestito riguardano tutti e tre gli attributi chiave visto che uno studente (MATRICOLA) prende in prestito un libro (ISBN e N\_COPIA) in un determinato giorno; per quanto riguarda le proroghe anch'esse sono relative all'intera chiave poiché il numero di proroghe fa riferimento ad un libro (ISBN e N\_COPIA) in prestito a uno specifico studente (MATRICOLA). La relazione SCRIVE non ha attributi al di fuori delle chiavi primarie. Nella relazione COPIA il dipartimento è funzionalmente dipendente dalla combinazione di ISBN e N\_COPIA poiché una specifica copia di un libro può essere in un dipartimento diverso da un'altra copia.
- La terza forma normale è stata anch'essa rispettata poiché è rispettata la seconda forma normale e non ci sono attributi non-chiave che dipendono funzionalmente da altri attributi non-chiave

## **PROBLEMA INIZIALE: Estrazione e popolamento del database**

Il primo problema riscontrato è stato avere un database popolato per poi eseguire le varie interrogazioni nel sito web. Inizialmente è stato volto lo sguardo al diagramma E/R e allo schema relazionale per la creazione delle tabelle nel database prestando particolare attenzione ai vincoli referenziali e alla loro gestione: Per visionare la creazione è disponibile il file CREAZIONE\_BIBLIOTECA.sql (O IL DUMP dell'intera base di dati) nella cartella "Creazione E Popolamento DB" all'indirizzo github sopra citato. Conseguentemente alla creazione sono stati estratti i dati (candidati all'estrazione) utilizzando i tool di import ed export in MySQLWorkbench: gli insert fatti sono disponibili nel file INSERT.sql nella stessa cartella del file precedente. Dopodiché sono stati utilizzati 3 programmi creati ad hoc per la generazione casuale di: STUDENTI, COPIE e PRESTITI

1. Il primo programma creato in java non è altro che un generatore casuale di Studenti che tramite degli array contenenti i vari attributi della relazione STUDENTE utilizza tali dati per la creazione casuale di 100 studenti con relative stringhe necessarie al popolamento.
2. Per generare le copie è stato utilizzato un programma PHP che tramite delle apposite query genera per ogni libro un numero di copie casuale che poi verranno associate a un dipartimento (scelto anch'esso casualmente per ogni copia). In fine sono state generate le stringhe necessarie agli inserimenti.
3. Per generare i prestiti è stato utilizzato lo stesso metodo sopra descritto in un altro programma PHP prestando però particolare attenzione ai vincoli: un libro (con uno specifico numero di copia e ISBN) già in prestito non può essere dato in prestito a nessun altro e per tale scopo è stato eseguito un ciclo di verifica oltre che un conseguente "INSERT" per non generare duplicati ed effettuare sempre il controllo ad ogni nuovo generamento.

## SITO:

Una volta popolato il database è stato creato il sito grazie anche all'utilizzo delle librerie Bootstrap (utile per la suddivisione corretta delle informazioni nella pagina e per l'abbellimento grafico) e jQuery (Utilizzata principalmente per funzioni di controllo dell'input e gestione di pulsanti). Il sito è stato suddiviso in 3 parti principali come già spiegato sopra le quali utilizzano il linguaggio PHP con SQL per recuperare, gestire e visualizzare le informazioni. Facendo un breve riassunto sulle funzionalità:

connect.php: una pagina contenente la funzione principale di connessione al database e la funzione get\_post() per il recupero dei dati dai form. Grazie alla funzione require() viene utilizzata poi nelle altre pagine.

Studente: In questa sezione troviamo l'inserimento, la modifica e l'eliminazione dal database delle informazioni di uno studente, tutte queste operazioni sono state ottenute grazie ad un form per l'inserimento dei dati. E' presente il controllo dell'input sui campi obbligatori, sulla matricola (tipo ed eventuale aggiunta o meno di zeri) e sulla presenza o meno di tale studente nel database per la modifica e l'eliminazione. Per la suddivisione dei form, ad esempio nell'aggiornamento dei dati, avviene prima l'inserimento per effettuare la ricerca nella database dei dati dello studente e poi l'eventuale modifica. Qui sono stati utilizzati dei campi di input, nascosti all'utente, per poi poter effettuare l'aggiornamento. L'eliminazione di uno studente viene negata nel caso in cui esso abbia dei prestiti ancora in corso come specificato nei vincoli.

Prestito: Qui sono gestiti i prestiti dei libri agli studenti: l'inserimento avviene tramite la registrazione delle informazioni necessarie nel database, è stata però presa la decisione di utilizzare solo la data corrente come data di inizio prestito poiché non avrebbe senso far cominciare un nuovo prestito nel passato o nel futuro. Viene inoltre inizializzato il numero di proroghe a 0 per ogni nuovo prestito che servirà poi nella sezione di modifica dati del prestito per il prolungamento di quest'ultimo. E' stata valutata come unica informazione utile da modificare il numero di proroghe (visto che la data di uscita viene inizializzata automaticamente): per calcolare la data prevista per la restituzione viene utilizzata la data di inizio prestito a cui verranno sommati 30 giorni. Nel caso in cui però sia stata fatta richiesta di un'eventuale proroga per il rientro verranno sommati ulteriori 15 giorni (per ogni proroga con un massimo di 2). Nell'eventuale caso di ritardi (verificabile nel tool di statistiche aggiuntive) il prestito viene mantenuto nel database e sarà poi compito della biblioteca decidere come procedere nei confronti dello studente. Infine è stato deciso che l'informazione sul prestito viene eliminata dal database al momento della restituzione. E' stato inoltre fornito un elenco completo con tutti i prestiti in corso.

Strumenti: Come richiesto sono state inserite 2 sezioni apposite: una per il recupero delle informazioni riguardanti i prestiti di uno specifico studente (o più di uno nel caso di omonimia) e una per il recupero delle informazioni riguardanti i prestiti di uno specifico libro, anche inserendo parzialmente il nome del libro o il nome dello studente. Per la visualizzazione delle statistiche di base è stata creata una pagina apposita che utilizza le query spiegate successivamente. Infine è stata creata una sezione apposita per le funzionalità aggiuntive nella quale si possono andare a recuperare altre informazioni utili alla gestione dell'intera biblioteca.

## INTERROGAZIONI BASE:

- "LE 5 LINGUE PIU FREQUENTI"



SQL:

```
SELECT LIBRO.NOME_LINGUA,COUNT(LIBRO.NOME_LINGUA) AS NUMERO_LIBRI
      FROM LIBRO
     GROUP BY LIBRO.NOME_LINGUA
    ORDER BY NUMERO_LIBRI DESC
    LIMIT 5;
```

ALGEBRA RELAZIONALE: per scegliere il numero di righe e quindi utilizzare il LIMIT in algebra relazionale in alcuni testi online viene fatto riferimento alla funzione ROWNUM() e infatti essa è stata utilizzata.

Per effettuare una rinominazione sul risultato del COUNT è stata utilizzata la sintassi riportata in seguito con la freccia indicante a destra.

```
TMP <-  $\pi$ _{NOME_LINGUA, NUMERO_LIBRI} (NOME_LINGUA \(\rightarrow\) COUNT(NOME_LINGUA)->NUMERO_LIBRI (LIBRO))
TMPORDERED <-  $\tau$ _{NUMERO_LIBRI DESC} (TMP)
RES <-  $\sigma$ _{ROWNUM() > 0 and ROWNUM() \(\leq\) 5} (TMPORDERED)
```

- **"AUTORE CHE HA SCRITTO PIU LIBRI"**

SQL: 2 query utilizzate, una interna per ottenere il massimo dei libri scritti dagli autori e una esterna per sfruttare tale numero in modo da recuperare le informazioni a essi associate,

```
SELECT A.ID_AUT,A.NOME_A,COGNOME_A,COUNT(S.ID_AUT) AS LIBRI_SCRITTI
FROM AUTORE A,SCRIVE S
WHERE A.ID_AUT=S.ID_AUT
GROUP BY A.ID_AUT
HAVING LIBRI_SCRITTI=(SELECT MAX(T.LIBRI_SCRITTI)
                      FROM(SELECT A.ID_AUT,A.NOME_A,COGNOME_A,COUNT(S.ID_AUT)
                               AS LIBRI_SCRITTI
                              FROM AUTORE A,SCRIVE S
                               WHERE A.ID_AUT=S.ID_AUT
                               GROUP BY A.ID_AUT
                               ORDER BY LIBRI_SCRITTI DESC) AS T)
ORDER BY LIBRI_SCRITTI DESC;
```

ALGEBRA RELAZIONALE:

```
TMP <- AUTORE * ( $\sigma$ _{AUTORE.ID_AUT = SCRIVE.ID_AUT} SCRIVE)
RES <-  $\pi$ _{ID_AUT, NOME_A, COGNOME_A, LIBRI_SCRITTI} (ID_AUT \(\rightarrow\) COUNT(ID_AUT)->LIBRI_SCRITTI (TMP))
MAX_VALUE <-  $\gamma$ _{MAX LIBRI_SCRITTI} (RES)
RESFINALE <-  $\sigma$ _{LIBRI_SCRITTI=MAX_VALUE} (RES)
```

- **"EDITORE CHE HA PUBBLICATO PIU LIBRI"**

SQL:

```
SELECT E.NOME_ED,COUNT(L.COD_ED) AS LIBRI_PUBBLICATI
```

```

FROM EDITORE E, LIBRO L
WHERE E.CODICE=L.COD_ED
GROUP BY L.COD_ED
HAVING LIBRI_PUBBLICATI=(SELECT MAX(T.LIBRI_PUBBLICATI)
                                FROM(SELECT E.NOME_ED,COUNT(L.COD_ED) AS
LIBRI_PUBBLICATI
                                FROM EDITORE E,LIBRO L
                                WHERE E.CODICE=L.COD_ED
                                GROUP BY L.COD_ED
                                ORDER BY LIBRI_PUBBLICATI DESC) AS T)

ORDER BY LIBRI_PUBBLICATI DESC;

```

## ALGEBRA RELAZIONALE:

```

TMP <- EDITORE * (EDITORE.CODICE = LIBRO.COD_ED) LIBRO
RES <-  $\pi$  NOME_ED, LIBRI_PUBBLICATI (COD_ED  $\gamma$  COUNT(COD_ED)->LIBRI_PUBBLICATI (TMP))
MAX_VALUE <-  $\gamma$  MAX LIBRI_PUBBLICATI (RES)
RESFINALE <-  $\sigma$  LIBRI_PUBBLICATI=MAX_VALUE (RES)

```

## INTERROGAZIONI AGGIUNTIVE:

- "INSERISCI NOME DEL LIBRO E TROVA IN QUANTE COPIE E' PRESENTE PER OGNI DIPARTIMENTO"

//qui è stata utilizzata una left join per simulare un'operazione di sottrazione

//\$titolo è il titolo che verrà inserito dall'utente

SQL:

```

SELECT C.NOME_DIP, COUNT(C.NUMERO_COPIA)
FROM (SELECT * FROM(SELECT ISBN,NUMERO_COPIA, NOME_DIP
                                FROM COPIA) X
LEFT JOIN
(
    SELECT P.ISBN,P.NUMERO_COPIA
    FROM PRESTITO P) T
ON X.ISBN = T.ISBN AND X.NUMERO_COPIA=T.NUMERO_COPIA
WHERE
    T.ISBN IS NULL AND T.NUMERO_COPIA IS NULL), LIBRO L
WHERE C.ISBN = L.ISBN AND L.TITOLO LIKE '$titolo'
GROUP BY L.TITOLO, C.NOME_DIP;

```

## ALGEBRA RELAZIONALE:

```

 $\rho$  T (  $\pi$  ISBN, NUMERO_COPIA (PRESTITO))
 $\rho$  X (  $\pi$  ISBN, NUMERO_COPIA, NOME_DIP (COPIA))
ResIntermedio <-  $X \bowtie (X.ISBN = T.ISBN \text{ AND } X.NUMERO\_COPIA = T.NUMERO\_COPIA)$  T
 $\rho$  C ( $\pi$  X.ISBN, X.NUMERO_COPIA, X.NOME_DIP ( $\sigma$  T.ISBN = NULL AND T.NUMERO_COPIA = NULL (ResIntermedio)))
//$titolo è il titolo del libro desiderato
TMP <- C * (C.ISBN = LIBRO.ISBN AND LIBRO.TITOLO = '$titolo') LIBRO
RES <-  $\pi$  C.NOME_DIP, COPIE_DISPONIBILI (TITOLO, NOME_DIP  $\gamma$ 
COUNT(NUMERO_COPIA)->COPIE_DISPONIBILI (TMP))

```

- **"STUDENTI CON PIU' PRESTITI ATTIVI"**

SQL:

```
SELECT S.MATRICOLA,S.NOME,S.COGNOME,COUNT(P.MATRICOLA) AS NUMERO_PRESTITI
FROM PRESTITO P,STUDENTE S
WHERE P.MATRICOLA=S.MATRICOLA
GROUP BY MATRICOLA
ORDER BY NUMERO_PRESTITI DESC
LIMIT 15;
```

ALGEBRA RELAZIONALE:

```
TMP <- PRESTITO *(PRESTITO.MATRICOLA = STUDENTE.MATRICOLA) STUDENTE
RES <- TT MATRICOLA, NOME, COGNOME, NUMERO_PRESTITI (MATRICOLA γ
COUNT(MATRICOLA)->NUMERO_PRESTITI (TMP))
RESFINALE <- τ NUMERO_PRESTITI DESC (RES)
```

- **"DIPARTIMENTI CON PIU' LIBRI"**

SQL:

```
SELECT D.NOME_DIP,COUNT(C.NOME_DIP) AS NUMERO_LIBRI
FROM DIPARTIMENTO D,COPIA C
WHERE D.NOME_DIP=C.NOME_DIP
GROUP BY NOME_DIP
ORDER BY NUMERO_LIBRI DESC;
```

ALGEBRA RELAZIONALE:

```
TMP <- DIPARTIMENTO *(DIPARTIMENTO.NOME_DIP = COPIA.NOME_DIP) COPIA
RES <- TT NOME_DIP, NUMERO_LIBRI (NOME_DIP γ COUNT(NOME_DIP)->NUMERO_LIBRI (TMP))
RESORDINATO <- τ NUMERO_LIBRI DESC (RES)
```

- **"INSERISCO LA DATA E TROVA I PRESTITI EFFETTUATI QUEL GIORNO"**

//CONTROLLO EFFETTUATO CON PHP SULLA BASE DEI DATI OTTENUTI

// %data si riferisce alla data inserita in input dall'utente per visualizzare i prestiti

//effettuati in quel giorno

SQL:

```
SELECT P.ISBN,L.TITOLO,P.NUMERO_COPIA,MATRICOLA,P.DATA_USCITA
FROM PRESTITO P,COPIA C,LIBRO L
WHERE P.ISBN=C.ISBN AND P.NUMERO_COPIA=C.NUMERO_COPIA AND C.ISBN=L.ISBN AND P.DATA_USCITA='%DATA';
```

ALGEBRA RELAZIONALE:

```

TMP <- PRESTITO *(PRESTITO.ISBN = COPIA.ISBN AND PRESTITO.NUMERO_COPIA = COPIA.NUMERO_COPIA)
COPIA
TMP2 <- TMP *(TMP.ISBN = LIBRO.ISBN AND DATA_USCITA = date('%data')) LIBRO
RES <- TT ISBN, TITOLO, NUMERO_COPIA, MATRICOLA, DATA_USCITA (TMP2)

```

- *“PRESTITI IN IMMINENTE SCADENZA”*

SQL:

```

SELECT S.MATRICOLA,S.NOME,S.COGNOME,P.ISBN,L.TITOLO,P.NUMERO_COPIA,P.DATA_USCITA, P.N_PROROGHE
FROM PRESTITO P,COPIA C,STUDENTE S,LIBRO L
WHERE P.ISBN=C.ISBN AND P.NUMERO_COPIA = C.NUMERO_COPIA AND P.MATRICOLA = S.MATRICOLA AND P.ISBN = L.ISBN;

```

ALGEBRA RELAZIONALE:

```

TMP <- PRESTITO *(PRESTITO.ISBN = COPIA.ISBN AND PRESTITO.NUMERO_COPIA = COPIA.NUMERO_COPIA)
COPIA
TMP2 <- TMP *(TMP.ISBN = LIBRO.ISBN) LIBRO
TMP3 <- TMP2 *(TMP2.MATRICOLA = STUDENTE.MATRICOLA) STUDENTE
RES <- TT MATRICOLA, NOME, COGNOME, ISBN, TITOLO, NUMERO_COPIA, DATA_USCITA, N_PROROGHE (TMP3)

```