

# Gestione Universitaria

## 1 Abstract

“Gestione universitaria” è un software che permette di gestire le biblioteche e i noleggi dei libri delle università nazionali ai fini di ottenere una migliore gestione delle proprie risorse, in modo da poter migliorare la qualità del servizio offerto.

Le università potranno venire a conoscenza di quanti libri i propri studenti comprano in un certo periodo di tempo, delle categorie più acquistate, dei bibliotecari che hanno prestato servizio all'interno delle varie biblioteche dell'università stessa, dei fornitori ed infine delle case editrici con cui collaborano.

Inoltre le università potranno tenere sotto controllo i loro fondi sia pubblici che privati per poter tenere traccia dei budget degli anni passati.

## 2 Analisi dei requisiti

### 2.1 Descrizione testuale

Nella base di dati sono presenti i dati degli studenti registrate alle università.

Di ogni **studente** sono noti i seguenti campi :

- Nome
- Cognome
- Sesso
- Numero\_di\_telefono
- Data\_di\_nascita
- Codice\_fiscale
- Matricola

Ad ogni studente iscritto ad un'università viene fornito un **badge**, di ogni badge è noto:

- Identificatore
- Codice\_a\_barre

La base di dati registra gli **autori** dei **libri**, in particolare di ogni **autore** è noto :

- Nome
- Cognome
- Sesso
- Codice\_fiscale
- Numero\_libri\_scritti
- Data\_di\_nascita

Per ogni **casa editrice**, sono noti :

- Nome
- Via
- Numero\_libri\_stampati
- Numero\_dipendenti
- Data\_di\_fondazione
- Partita\_IVA

Per ogni **libro** scritto da un **autore** e pubblicato da una **casa editrice**, sono noti :

- Titolo
- Numero\_pagine
- Numero\_capitoli
- Numero\_copie
- Genere
- Eta\_consigliata
- Incassi\_totali

- Prezzo\_unitario
- Punteggio\_medio

I **libri** prodotti dalle **case editrici** vengono distribuiti dai fornitori, di ogni **fornitore** è noto :

- Nome
- Partita\_IVA
- Via\_magazzino
- Numero\_furgoni

I libri contenuti all'interno di una biblioteca sono riposti dentro a degli scaffali, di ogni **scaffale** è noto :

- Genere
- Numero\_ripiani
- Codice\_scaffale

La base di dati salva le **biblioteche** di ogni università, per ogni **biblioteca** è noto :

- Nome
- Numero\_posti
- Numero\_libri\_contenuti
- Numero\_prese\_corrente
- Presenza\_distributore

Ogni **biblioteca** viene gestita da un **bibliotecario**, di ogni bibliotecario è noto :

- Nome
- Cognome
- Codice fiscale
- Data\_di\_nascita
- Sesso
- Stipendio

Delle **università** salvate nella base di dati è noto :

- Nome
- Via
- Codice\_fiscale
- Numero\_studenti
- Data\_fondazione
- Tassa\_annuale
- Cinque\_per\_mille

I **Noleggi** dei libri vengono effettuati da **studenti** nelle **biblioteche** delle loro università, di ogni noleggio è noto :

- ID\_Noleggio
- Data\_Inizio
- Data\_Fine
- Importo\_Sanzione
- Motivo (Della Sanzione)
- Punteggio\_Stelle

Del **Fondo** di ogni università è noto:

- Numero\_Finanziamento
- Data\_Finanziamento
- Importo

## 2.2 Glossario dei termini

Termine	Descrizione	Collegamenti
Studente	Identifica uno studente regolarmente iscritto ad un istituto universitario.	Libro, Badge, Università
Badge	Rappresenta un badge universitario che abilita l'accesso alle biblioteche per poter prendere in noleggio uno o più libri.	Studente, Università
Autore	Rappresenta una persona che di professione svolge l'attività di scrittore.	Libro
Libro	Rappresenta un libro scritto da un unico autore.	Casa editrice, Autore, Studente, Scaffale
Casa Editrice	Identifica la casa editrice che ha revisionato e stampato i libri prodotti dai rispettivi autori.	Libro, Fornitore
Fornitore	Identifica un fornitore che fornisce le biblioteche degli istituti prelevando i libri direttamente dalle case editrici.	Casa editrice, Biblioteca
Biblioteca	Identifica una biblioteca all'interno di un istituto universitario.	Libro, Fornitore, Bibliotecario
Università	Identifica un istituto universitario nazionale.	Studente, Biblioteca, Badge, Fondo,
Fondo	Rappresenta un contributo economico alle università.	Università
Scaffale	Rappresenta uno scaffale situato all'interno di una biblioteca.	Libro, Biblioteca
Bibliotecario	Identifica il dipendente che si occupa di gestire la biblioteca.	Biblioteca
Matricola	Rappresenta il numero associato a ciascun studente all'interno del proprio istituto universitario.	Nessuno
Punteggio medio	Indica il punteggio medio per un determinato libro.	Nessuno

### 2.3 Operazioni

OPERAZIONE	TIPO	FREQUENZA
Inserimento di un nuovo libro	S	100 al giorno
Inserimento di un nuovo fornitore	S	1000 al giorno
Inserimento di un nuovo acquisto	S	100 al giorno
Ricerca del libro maggiormente venduto	L	20000 al giorno
Ricerca del libro meno venduto	L	10000 al giorno
Ricerca del libro che ha generato il maggior profitto.	L	10000 al giorno
Ricerca del libro che ha venduto il maggior numero di copie	L	15000 al giorno
Ricerca dei libri che hanno venduto almeno 100 copie.	L	20000 al giorno
Visualizzare lo studente che ha speso di più in un singolo importo.	L	10000 al giorno
Visualizzare gli studenti che hanno speso un importo maggiore di 40 euro.	L	15000 al giorno
Visualizzare il punteggio medio di ogni libro	L	2000 al giorno
Visualizzare il libro che ha il maggior punteggio medio	L	1500 al giorno

## 3 Progettazione Concettuale

### 3.1 Lista entità

Se non specificato l'attributo è NOT NULL

- **Studente:**
  - Nome: varchar(50)
  - Cognome: varchar(50)
  - Sesso: char(7)
  - Numero di telefono: char(9)
  - Codice fiscale: char(16)
  - Data di nascita: date
  - Matricola: integer primary key
- **Autore:**
  - Codice fiscale: char(16) primary key
  - Nome: varchar(50)
  - Cognome: varchar(50)
  - Sesso: char(7)
  - Data\_di\_nascita: date
  - Numero\_libri\_scritti: smallint
- **Badge:**
  - ID: smallint primary key
  - Codice\_barre: char(15)
- **Università:**
  - Nome: varchar(50) primary key
  - Via: varchar(50)
  - Codice\_fiscale: char(16)
  - Numero\_studenti: int
  - Tassa\_annuale: smallint

- Data\_fondazione: date
  - Cinque\_per\_mille character(10)
- **Casa\_Editrice:**
  - Nome: varchar(50)
  - Via: varchar(50)
  - Numero\_dipendenti: smallint
  - Numero\_libri\_stampati: int
  - Data\_fondazione: date
  - Partita IVA: char(11) primary key
- **Fornitore:**
  - Nome: varchar(50)
  - Via: varchar(50)
  - Partita IVA: char(11) primary key
  - Numero\_furgoni: smallint
- **Biblioteca:**
  - Nome: varchar(50) primary key
  - Numero\_posti: smallint
  - Numero\_libri\_contenuti: smallint
  - Prese\_corrente: smallint
  - Presenza\_distributore: bool
- **Scaffale:**
  - Codice: smallint primary key
  - Genere: varchar(50)
  - Numero\_ripiani: smallint
- **Libro:**
  - Titolo: varchar(100) primary key
  - Numero\_pagine: smallint
  - Numero\_capitoli: smallint
  - Genere: varchar(50)
  - Eta\_consigliata: smallint
  - Prezzo\_unitario: decimal(3,1)
  - Numero\_copie\_disponibili: smallint
  - Incassi\_totali: double
  - Punteggio\_medio: decimal(2,1), >=0, <= 5
- **Bibliotecario:**
  - Codice\_fiscale: char(16) primary key
  - Nome: varchar(50)
  - Cognome: varchar(50)
  - Data\_nascita: date
  - Sesso: char(7)
  - Eta: smallint
  - Stipendio: smallint
- **Fondo:**
  - Data\_versamento: date primary key
  - Importo: integer
  - Numero\_Finanziamento: integer

L'entità **Fondo** si specializza in due sottocategorie con una generalizzazione totale:

- Fondo\_Pubblico
  - Costi\_previsti: integer, costi\_previsti<importo

- Fondo\_Privato di cui sono noti i seguenti campi :
  - Nome Benefattore: varchar(50)
  - Cognome Benefattore: varchar(50)
- Noleggio:
  - ID\_Noleggio: integer primary key
  - Data\_Inizio: date
  - Data\_Fine: date
  - Punteggio\_Stelle: smallint, 1>= e 5<=
  - Importo\_Sanzione: integer, 50> e 200<
  - Motivazione: varchar(100)

### 3.2 Tabella delle relazioni

Relazione	Entità Coinvolte	Descrizione	Attributi
Scrive	Autore(1,N) Libro(1,1)	Un autore può aver scritto più di un libro, un libro è scritto da un solo autore.	Nessuno
Stampa	CasaEditrice(1,N) Libro(1,1)	Una casaEditrice stampa più libri, un libro è stampato da una sola casaEditrice.	Nessuno
fornisce_da	CasaEditrice(1,N) Fornitore(1,N)	Una casa Editrice fornisce più fornitori, più fornitori possono rifornirsi dalla stessa casaEditrice.	Nessuno
fornisce_a	Fornitore(1,N) Biblioteca(1,N)	Un fornitore fornisce una o più biblioteche, una biblioteca è fornita da uno o più fornitori.	Nessuno
Composta	Biblioteca(1,N) Scaffale(1,1)	Una biblioteca è composta da più scaffali, uno scaffale è identificato in maniera univoca all'interno della biblioteca.	Nessuno
Contiene	Libro(1,N) Scaffale(1,N)	Una stessa copia di un libro, può risiedere in più scaffali purché siano dello stesso genere.	Nessuno
Possiede	Studente(1,1) Badge(1,1)	Uno studente deve possedere uno e un solo badge, un badge è associato ad un solo studente.	Nessuno
Lavorato_a	Bibliotecario(0,N) Biblioteca(1,N)	Un bibliotecario può aver lavorato in altre biblioteche, una biblioteca può aver avuto più bibliotecari.	Data_inizio, Data_fine
Lavora	Bibliotecario(1,1) Biblioteca(1,1)	Un bibliotecario può lavorare attualmente in una sola biblioteca, una biblioteca può avere un solo bibliotecario attivo.	Data_inizio

Ottiene	Università(1,N) Fondo(1,1)	Un'università riceve ogni anno un importo da spendere, un importo da spendere è associato in maniera univoca ad un'università.	Nessuno
Frequenta	Studente(1,1) Università(1,N)	Uno studente studia in una sola università, un'università ha più studenti.	Nessuno
Acquisto	Studente(0,N) Libro(0,N)	Uno studente può aver effettuato uno o più acquisti di un libro, un libro può essere stato acquistato zero o più volte. Ogni studente può comprare uno stesso libro una sola volta al giorno.	Data_acquisto, Importo, Numero_copie
Produzione	Università(1,N) Badge(1,1)	Ogni badge è prodotto da una e una sola università. Ogni università produce più badge, uno per ogni studente.	Nessuno
Appartiene	Università(1,N) Biblioteca(1,1)	Una biblioteca appartiene ad una sola università (visto il nome), un'università ha più di una biblioteca.	Nessuno
Di	Libro(0,N) Noleggio(1,1)	Un libro può essere noleggiato più volte, in un noleggio si può noleggiare solo un libro.	Nessuno
Presso	Biblioteca(0,N) Noleggio(1,1)	Un noleggio può essere fatto presso una biblioteca, una biblioteca può fare più noleggi.	Nessuno
Fare	Studente(0,N) Noleggio(1,1)	Uno studente può fare più noleggi e un noleggio è fatto da un solo studente.	Nessuno

#### Vincoli non rappresentabili tramite schema E-R:

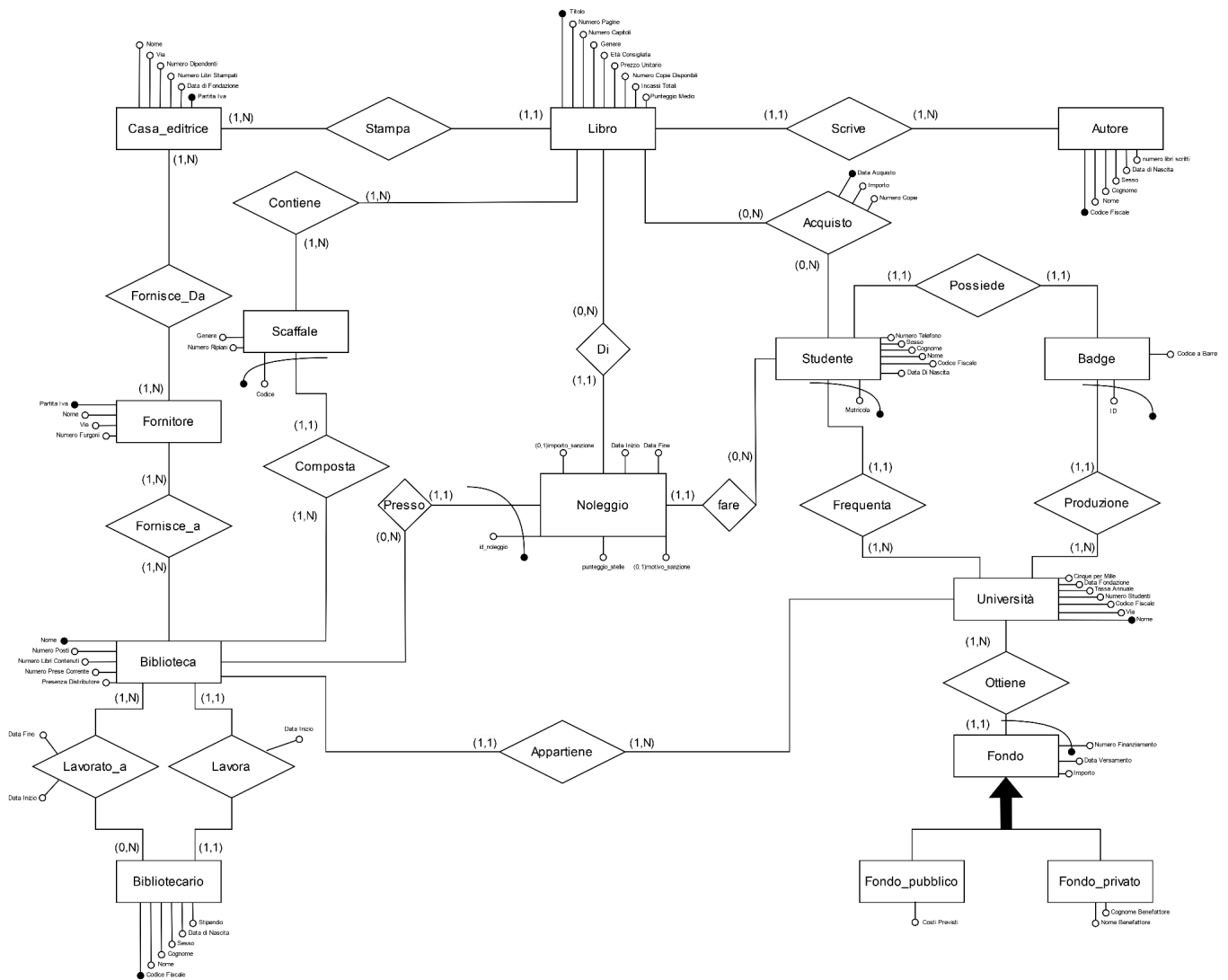
- Non è possibile specificare che l'importo assegnato alle università ogni anno è in milioni di euro.
- Non è possibile specificare che l'importo di un'eventuale sanzione è un valore intero costante maggiore di 50 e inferiore a 200 euro.
- Non è possibile specificare che il numero di stelle assegnate ad ogni noleggio è un numero  $\geq$  ad 1 e  $\leq$  a 5.
- Non è possibile specificare che il numero di caratteri per il motivo della sanzione è di 100 caratteri.
- Non è possibile specificare che la data di fine lavoro presso una biblioteca viene dopo la data di inizio.
- Non è possibile specificare che la data di inizio e di fine contratto di un bibliotecario presso una biblioteca non può andare a sovrapporsi con un altro periodo presso la stessa oppure un'altra biblioteca.
- Non è possibile specificare che l'importo (statale) assegnato alle università è sempre strettamente maggiore dei costi\_stimati.

- Non è possibile specificare che i costi\_previsti sono espressi in milioni di euro.

### Vincoli di derivazione:

- Il valore incassi totale di un libro è uguale alla somma dei singoli acquisti.
- Il valore di stelle medio per un libro è uguale alla media delle stelle assegnategli per ogni noleggio.

### Schema E-R Concettuale





## 4. Progettazione Logica

### 4.1 Ristrutturazione

#### 4.1.1 Analisi delle ridondanze

L'attributo "**Punteggio\_medio**" dell'entità Libro è ridondante in quanto può essere calcolato facendo la media dei punteggi dei noleggi relativi al libro.

Concetto	Tipo	Volume
Libro	Entità	250000
Noleggio	Entità	5000
Studente	Entità	250000

- OPERAZIONE 1: Inserire un nuovo noleggio con il relativo libro noleggiato (2000 volte al giorno)
- OPERAZIONE 2: Visualizzare il libro con il maggior punteggio medio. (1500 volte al giorno)

#### PRESENZA RIDONDANZA:

Operazione 1:

Un libro ha in media  $250000 / 5000 = 50$  noleggi.

Concetto	Costrutto	Accessi	Tipo
Libro	Entità	1	S
Noleggio	Entità	1	S
Di	Relazione	1	L

X 2000 volte al giorno

X 2000 volte al giorno

X 2000 volte al giorno

Operazione 2:

Concetto	Costrutto	Accessi	Tipo
Libro	Entità	250000	L

X 1500 volte al giorno

considerando che gli accessi in scrittura costano il doppio di quelli in lettura otteniamo un costo giornaliero pari a 375208000 dato dal calcolo  $= 2000*2*50+2000*2+2000+2000+250000*1500$ .

## ASSENZA RIDONDANZA:

Operazione 1:

Concetto	Costrutto	Accessi	Tipo
Libro	Entità	1	L
Noleggio	Entità	1	S
Di	Relazione	1	L

X 2000 volte al giorno

X 2000 volte al giorno

X 2000 volte al giorno

Operazione 2:

Concetto	Costrutto	Accessi	Tipo
Libro	Entità	250000	L
Di	Relazione	250000*50	L
Noleggio	Entità	250000*50	L

X 1500 volte al giorno

X 1500 volte al giorno

X 1500 volte al giorno

$$2000 + 2000 + 2000 + 250000 \cdot 1500 + 250000 \cdot 1500 \cdot 50 + 250000 \cdot 1500 \cdot 50 = 37875006000$$

In conclusione l'analisi della ridondanza dimostra che conviene mantenere l'attributo **"Punteggio\_Medio"** che verrà aggiornato con l'aggiunta di ogni noleggio.

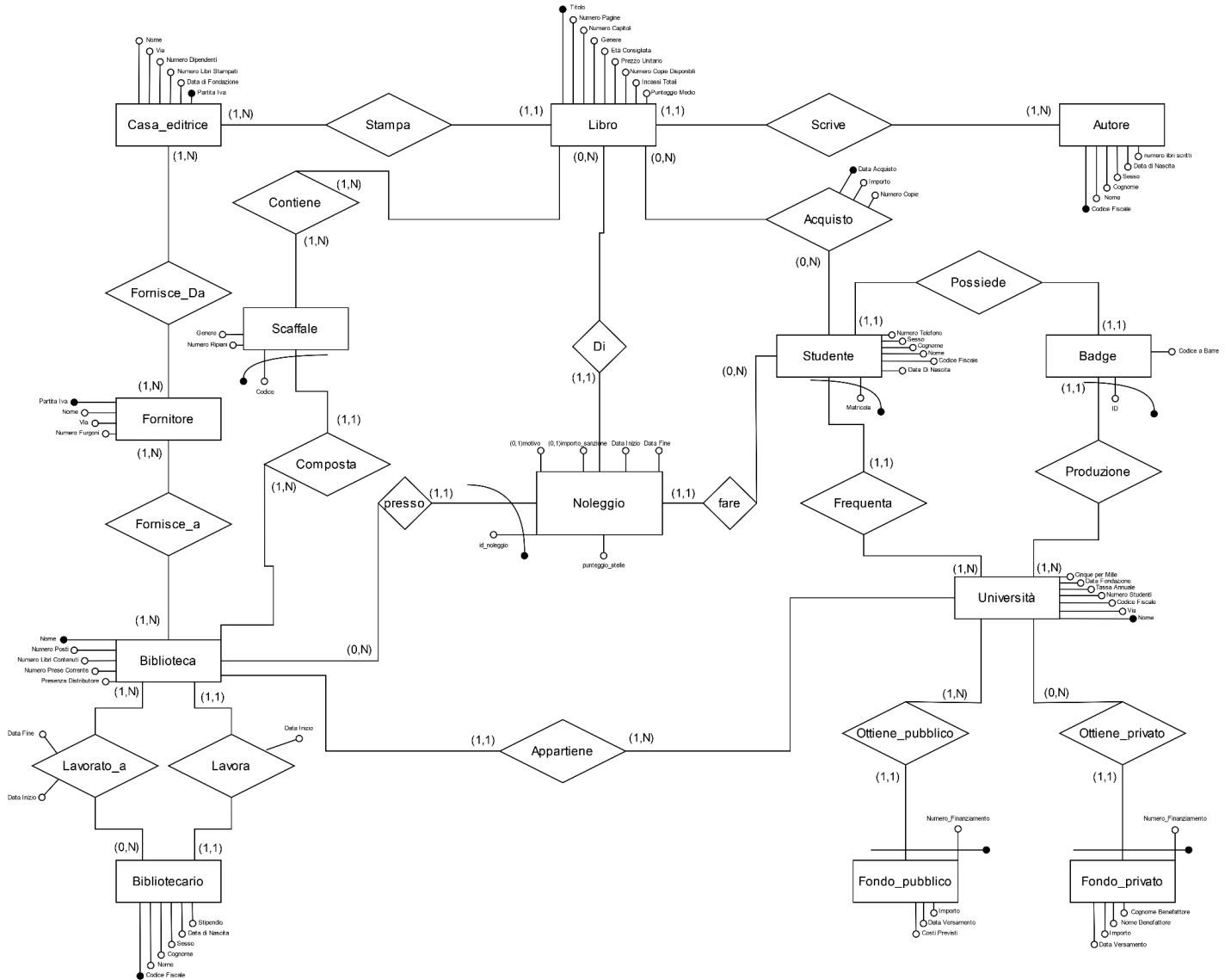
### 4.1.2 Eliminazione delle generalizzazioni

#### 4.1.2.1 Legenda

- $\leq$  (indica una generalizzazione **parziale**)
- $\leftarrow$  (indica una generalizzazione **totale**)

Generalizzazione	Risoluzione
Fondo $\leftarrow$ Fondo_pubblico, Fondo_privato	L'entità Fondo viene accorpata nelle entità figlie, questo perché gli accessi alle due entità sono contestuali, inoltre per quanto riguarda l'entità fondo_privato vengono aggiunti il nome e il cognome della persona che ha effettuato la donazione, invece per l'entità fondo_pubblico andiamo ad aggiungere il costo previsto dall'università in quell'anno.

## Schema E-R ristrutturato



### 4.2 Creazione delle tabelle (A→B indica che B è chiave esterna di A)

**Autore** ( codice\_fiscale, nome, cognome, sesso, data\_nascita, libri\_scritti)

**Bibliotecario** ( codice\_fiscale, nome, cognome, sesso, data\_nascita, stipendio, data\_inizio)

**Casa Editrice** (nome, via, numero\_dipendenti, numero\_libri\_stampati, data\_fondazione, partita\_iva)

**Libro** ( titolo, numero\_pagine, numero\_capitoli, genere, eta\_consigliata, prezzo\_unitario, numero\_copie\_disponibili, incassi\_totali, punteggio\_medio, editrice->casa\_editrice.partita\_iva, autore->autore.codice\_fiscale)

**Università** ( nome, via, codice\_fiscale, numero\_studenti, tassa\_annuale, data\_fondazione, cinque\_per\_mille)

**Biblioteca** ( nome, numero\_posti, numero\_libri\_contenuti, numero\_prese\_corrente, presenza\_distributore, bibliotecario->bibliotecario.codice\_fiscale, universita->universita.nome)

**Scaffale** ( codice, biblioteca->biblioteca.nome, genere, numero\_ripiani)

**Badge** ( id, codice\_barre, universita->universita.nome)

**Studente** ( nome, cognome, codice\_fiscale, sesso, numero\_telefono, data\_nascita, matricola, universita->universita.nome, badge->badge.id)

**Contiene** ( libro->libro.titolo, scaffale->scaffale.codice, biblioteca->scaffale.biblioteca)

**Fondo Pubblico** ( numero\_finanziamento, data\_versamento, importo, costi\_previsti, universita->universita.nome)

**Fondo Privato** ( numero\_finanziamento, data\_versamento, importo, universita->universita.nome, nome\_benefattore, cognome\_benefattore)

**Fornitore** ( partita\_iva, nome, via, numero\_furgoni)

**Fornisce A** ( biblioteca->biblioteca.nome, p\_iva\_fornitore->fornitore.partita\_iva)

**Fornisce Da** ( casa\_editrice->casa\_editrice.partita\_iva, p\_iva\_fornitore->fornitore.partita\_iva)

**Lavorato A** ( biblioteca->biblioteca.nome, bibliotecario->bibliotecario.codice\_fiscale, data\_inizio, data\_fine)

**Noleggio** ( id, data\_inizio, data\_fine, libro->libro.titolo, matricola->studente.matricola, universita->universita.nome, punteggio\_stelle, biblioteca->biblioteca.nome, importo\_sanzione, motivazione)

**Acquisto** ( data\_acquisto, importo, numero\_copie, libro->libro.titolo, matricola->studente.matricola, universita->universita.nome)

## 5 Query e Indici

### 5.1 Query

1. Ricerca dell'autore che ha avuto più incassi.

```
SELECT a.nome, a.cognome, SUM(l.incassi_totali) FROM autore AS a
INNER JOIN libro as l ON l.autore = a.codice_fiscale
GROUP BY a.nome,a.cognome
ORDER BY SUM(l.incassi_totali) DESC
```

2. Visualizzare per ogni noleggio con una sanzione maggiore di 20 euro il motivo.

```
SELECT n.id, n.motivazione FROM noleggio AS n
WHERE (n.importo_sanzione > 100)
```

3. Per ogni università visualizzare i badge degli studenti che hanno effettuato degli acquisti con un importo maggiore di 100 euro.

```
SELECT b.codice_barre, s.nome, s.cognome FROM badge AS b
INNER JOIN studente as s ON b.id=s.badge AND b.universita=s.universita
INNER JOIN acquisto as a ON s.matricola=a.matricola AND s.universita=a.universita
WHERE (a.importo>20)
GROUP BY b.codice_barre, s.nome, s.cognome
```

4. Per ogni università visualizzare (per categorie) i libri venduti con un costo maggiore di 30 euro.

```
SELECT u.nome, l.genere, a.importo FROM universita AS u
INNER JOIN studente AS s ON u.nome = s.universita
INNER JOIN acquisto AS a ON u.nome = a.universita AND s.matricola=a.matricola
INNER JOIN libro AS l ON a.libro = l.titolo
WHERE a.importo>30
ORDER BY u.nome, l.genere
```

5. Visualizzare i libri in ordine di punteggio medio crescente.

```
SELECT l.titolo, l.punteggio_medio FROM libro AS l
ORDER BY l.punteggio_medio ASC
```

6. Per ogni università visualizzare i bibliotecari che ci hanno lavorato.

```
SELECT u.nome, b.nome, b.cognome FROM universita AS u
INNER JOIN biblioteca AS bib ON u.nome = bib.universita
INNER JOIN lavorato_a AS la ON la.biblioteca = bib.nome
INNER JOIN bibliotecario AS b on b.codice_fiscale = la.bibliotecario
ORDER BY u.nome
```

7. Visualizzare la Casa editrice che fornisce più biblioteche.

```
SELECT ce.nome, COUNT(DISTINCT bib.nome) FROM casa_editrice AS ce
INNER JOIN fornisce_da AS fd ON ce.partita_iva = fd.casa_editrice
INNER JOIN fornitore AS f ON fd.p_iva_fornitore = f.partita_iva
INNER JOIN fornisce_a AS fa ON f.partita_iva = fa.p_iva_fornitore
INNER JOIN biblioteca AS bib ON fa.biblioteca = bib.nome
GROUP BY ce.nome
ORDER BY COUNT(DISTINCT bib.nome) DESC
```

8. Visualizzare le Università in ordine di numero di libri contenuti nelle sue biblioteche.

```
SELECT u.nome, b.numero_libri_contenuti FROM universita AS u
INNER JOIN biblioteca AS b ON u.nome=b.universita
GROUP BY u.nome, b.numero_libri_contenuti
ORDER BY b.numero_libri_contenuti DESC
```

9. Visualizzare i fondi medi di ogni università.

```
SELECT u.nome, AVG(fpub.importo + fpri.importo) FROM universita AS u
INNER JOIN fondo_pubblico AS fpub ON u.nome=fpub.universita
INNER JOIN fondo_privato AS fpri ON u.nome=fpri.universita
GROUP BY u.nome
```

10. Per ogni biblioteca visualizzare quelle che contengono più di 20 libri.

```
SELECT b.nome, COUNT(c.libro) FROM biblioteca AS b
INNER JOIN scaffale AS s ON b.nome = s.biblioteca
INNER JOIN contiene AS c ON s.biblioteca = c.biblioteca AND s.codice=c.scaffale
GROUP BY b.nome
HAVING (COUNT(c.libro)>20)
```

## 5.2 Indice Secondario

L'aggiunta e la modifica di nuove biblioteche è poco frequente (circa 400 operazioni all'anno).

Invece la visualizzazione delle biblioteche per controllare l'università di appartenenza è molto frequente (circa 1000 operazioni ogni giorno), quindi richiede una certa velocità di ricerca tra tutte le biblioteche del territorio.

Quindi da questi riscontri e presupponendo uno sviluppo su larga scala di tutte le biblioteche universitarie nazionali possiamo indicizzare gli attributi Nome e Università nell'indice bib\_uni.

```
CREATE INDEX bib_unit ON biblioteca(nome, universita)
```

## 6 Codice C++

### 6.1 Descrizione dell'utilizzo del codice

Il codice C++ per l'esecuzione delle query consiste in un unico file .cpp, che va compilato attraverso il comando `g++ Client.cpp -L dependencies\lib -lpq -o client -std=c++11`.

Prima di poter compilare il file è necessario copiare i file libpq.dll e libpq.lib in `./dependencies/lib` e copiare

libpq-fe.h, pg\_consigt\_ext.h e postgres\_ext.h in `./dependencies/include`, dove `./` è il percorso della directory che contiene il file .cpp.

Per eseguire il codice, basta avviare l'eseguibile "Client" da CLI.

All'avvio del programma viene visualizzato un menu che specifica le query che si possono eseguire.

```
PS W:\Desktop\ProgettoBasi> ./client
Connessione in corso...
Connesso
Seleziona la query da eseguire:
0. Esci
1. Ricerca dell'autore che ha avuto più incassi.
2. Visualizzare per ogni noleggio con una sanzione maggiore di 20 euro il motivo.
3. Per ogni università visualizzare i badge degli studenti che hanno effettuato degli acquisti con un importo maggiore di 100 euro.
4. Per ogni università visualizzare (per categorie) i libri venduti con un costo maggiore di 30 euro.
5. Visualizzare i libri in ordine di punteggio medio crescente.
6. Per ogni università visualizzare i bibliotecari che ci hanno lavorato.
7. Visualizzare la Casa editrice che fornisce più biblioteche.
8. Visualizzare le Università in ordine di numero di libri contenuti nelle sue biblioteche.
9. Visualizzare i fondi medi di ogni università.
10. Per ogni biblioteca visualizzare quelle che contengono più di 20 libri.
```

### 6.2 Documentazione del codice

Definizione delle costanti per la connessione al database Postgres.

```
#define NOMEDB "Biblioteca"
#define HOST "127.0.0.1"
#define USER "postgres"
#define PASS "postgres"
#define PORT 5432
```

**Funzioni:**

```
void printQuery(PGresult* result)
```

Funzione che stampa il risultato della query nel terminale formattando la risposta del Database in tabella.

```
PGresult* eseguiQuery(PGconn* conn, const char* query)
```

Funzione che esegue la query tramite la connessione al database e la stringa query.

Esempio di esecuzione di alcune query :

```
/*1 Ricerca dell'autore che ha avuto più incassi.*/
"SELECT a.nome, a.cognome, SUM(l.incassi_totali) FROM autore AS a "
"INNER JOIN libro as l ON l.autore = a.codice_fiscale "
"GROUP BY a.nome,a.cognome "
"ORDER BY SUM(l.incassi_totali) DESC",
```

1

Risultato:

nome	cognome	sum
Arianna	Parsi	39108
Roberto	Tommasi	36796
Domenico	Pasquale	27489
Giorgio	Orvello	20790
Giovanna	Caruso	2379

Premi Enter|