

Relazione Progetto Basi dati

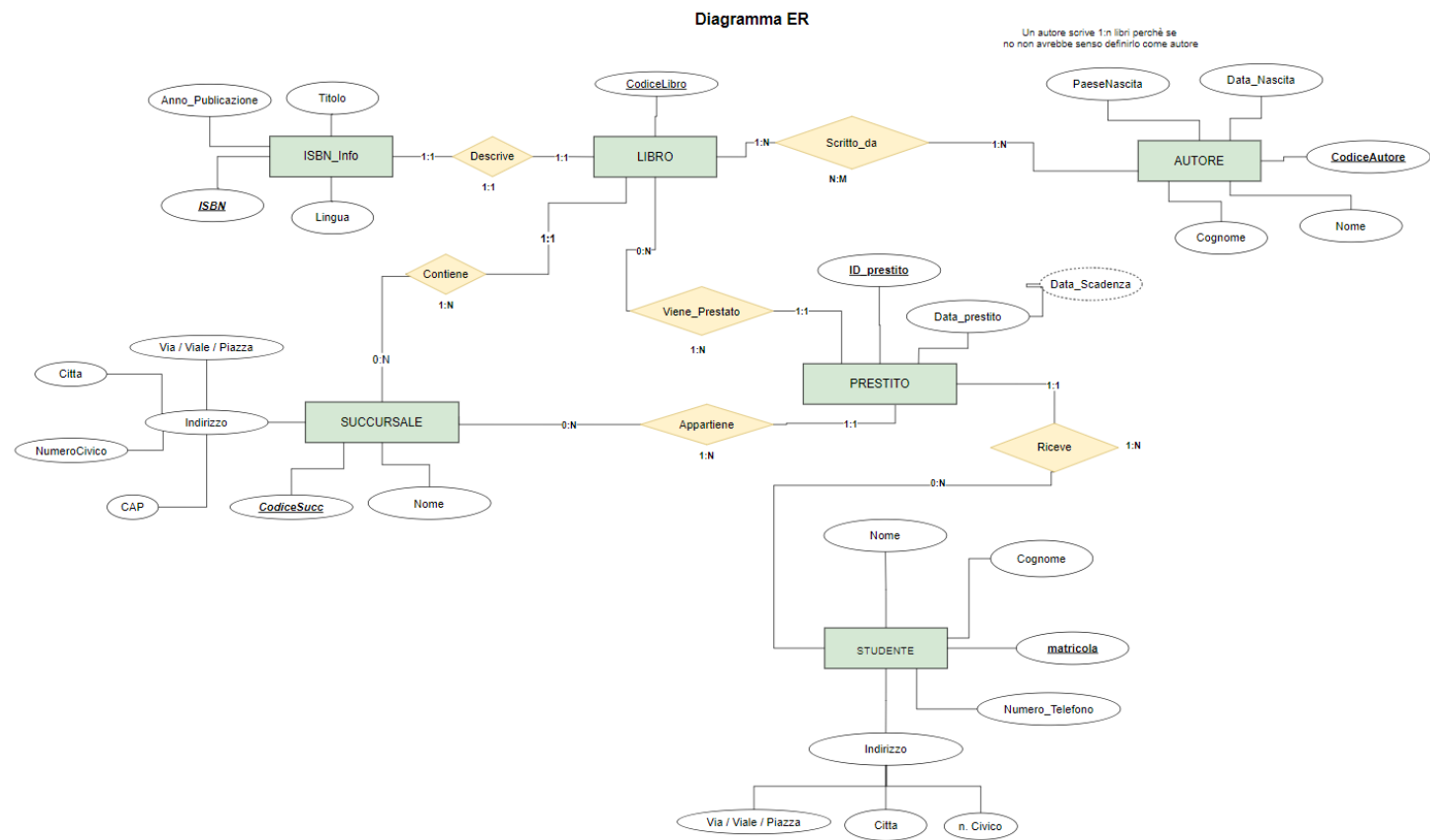
Edoardo Ponsanesi [166205]

Enrico Albertini [165672]

1. Definizione del problema

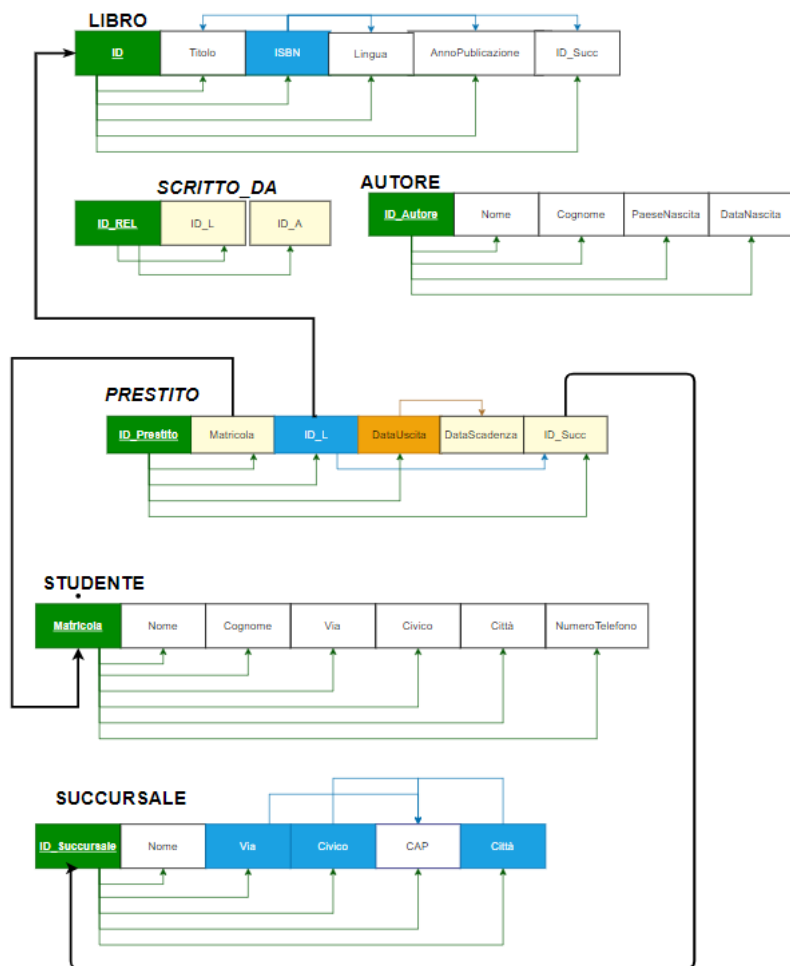
Ci è stata commissionata la costruzione di un sistema di gestione di una biblioteca, formata da una serie di succursali dell' università di ferrara. Ciò è stato implementato in html/php per l'interfaccia web dinamica che dialoga con un database MySQL contenente i dati.

2. Modello ER



3. Modello relazionale in terza forma normale

Schema PRENORMALIZZAZIONE



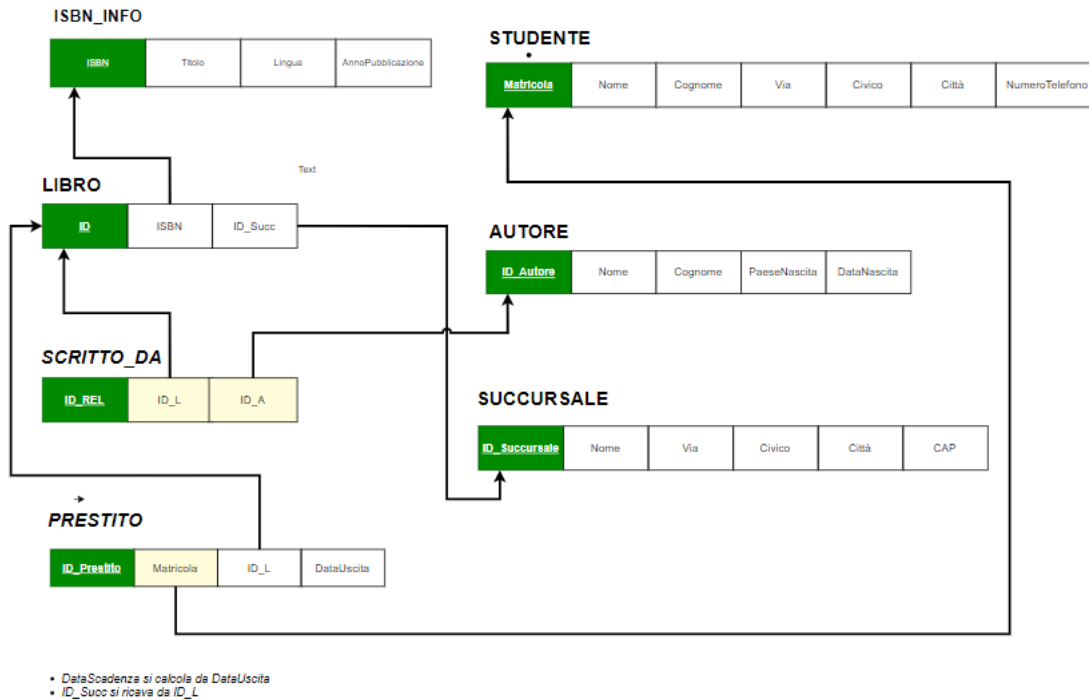
Abbiamo ritenuto opportuno riportare lo schema che sarebbe emerso naturalmente da un'osservazione non articolata dal punto di vista dei database, quindi non al corrente dei rischi di design legati alle dipendenze funzionali. LE DIPENDENZE FUNZIONALI SONO MOSTRATE SECONDO I COLORI DEL DIAGRAMMA, EVITIAMO DI RIPETERLE IN FORMA TESTUALE QUA PER EVITARE INUTILI RIDONDANZE.

Bisogna notare che:

- In PRESTITO:
DataScadenza è T.F.D.(Transitivamente Funzionalmente Dipendente) da ID_Prestito, viene inclusa nello schema perché è concettualmente un campo di prestito, ma nella realtà non viola la terza forma normale perché non è un campo effettivo, ma viene calcolato al bisogno da DataUscita(unico dei due mantenuto effettivamente nel DB)

Schema 3NF

FROM ISBN_Info AS info, Libro as l, Succursale as succ
WHERE info.TITOLO LIKE '%\$nome_libro%' AND succ.ID_SUCC = l.ID_S;



Abbiamo dunque rimosso tutte le **D.F.T.** (*Dipendenze Funzionali Transitive*), ottenendo lo schema in **3NF**.

A eccezione del CAP (funzionalmente dipendente da Via, Civico, Città), in un contesto implementativo generale), QUESTO E' VOLUTO COME SCELTA DI DESIGN, IN FUNZIONE DEL DOMINIO DI APPLICAZIONE DEL PRODOTTO RICHIESTO. Infatti realizzare schema, compliant alla non-DFT del CAP, avrebbe implicato di avere un database con tutte le vie della città di Ferrara, cosa che non è stata fornita. Quindi tecnicamente il CAP è solo un'altro attributo che va a costituire l'attributo composto dell'indirizzo con CAP.

4. Interrogazioni delle tracce in SQL con l' equivalente espressione scritta in Algebra Relazionale

[1]

Ricerca di un libro inserendo il titolo (anche parziale) - nel caso in cui nessun parametro venga specificato deve essere presentata la lista completa dei libri comprese le informazioni sintetiche del libro: titolo, isbn, in che succursale sono, ecc... (sintetiche - nome, cognome) sull'autore.

```
SELECT i.ISBN, i.TITOLO, i.LINGUA, s.NOME
FROM ISBN_Info AS i, Libro as l, Succursale as s
WHERE i.ISBN = l.ISBN AND i.TITOLO LIKE '%$nome_libro%'
AND l.ID_S = s.ID_SUCC
```

$\rho_i(\text{ISBN_Info}), \rho_l(\text{Libro}), \rho_s(\text{Succursale})$

$OUT \leftarrow \pi_{(ISBN, TITOLO, LINGUA, NOME)}(ISBN_Info \bowtie_{<i.ISBN=l.ISBN \wedge i.TITOLO=\%nome_libro\%>} Libro \bowtie_{<l.ID_S=s.ID_SUCC>} Succursale)$

[2] Visualizzazione di tutti i libri di un determinato autore, eventualmente suddivisi per anno di pubblicazione.

```
SELECT i.TITOLO, i.ANNO_PUBBLICAZIONE, i.LINGUA, l.ISBN
FROM Libro AS l, ISBN_Info AS i
WHERE l.ISBN = i.ISBN
AND l.ID_LIBRO IN ( SELECT sd.ID_L FROM Scritto_Da AS sd WHERE sd.ID_A = $id_autore)
ORDER BY i.ANNO_PUBBLICAZIONE";
```

$$\begin{aligned}
 & \rho_l(Libro), \rho_i(ISBN_Info), \rho_{sd}(Scritto_Da) \\
 DA_CERCARE & \leftarrow \pi_{<sd.ID_L>(\sigma_{<sd.ID_A=id_autore>(Scritto_Da))} \\
 & \rho_{dc}(DA_CERCARE) \\
 & - \\
 INFO_LIBRI & \leftarrow ISBN_Info \bowtie_{<l.ISBN=i.ISBN>} (\sigma_{<l.ID_LIBRO \in dc.ID_L>}(DA_CERCARE)) \\
 OUT & \leftarrow \pi_{<i.TITOLO, i.ANNO_PUBBLICAZIONE, i.LINGUA, i.ISBN>} (INFO_LIBRI)
 \end{aligned}$$

[3]

Ricerca degli autori inserendo uno o più parametri (anche parziali), in forma libera o eventualmente guidata (per esempio menù a tendina con i soli valori possibili).

```
SELECT NOME, COGNOME, ID_AUTORE, DATA_NASCITA, PAESE_NASCITA
FROM Autore
WHERE NOME LIKE '%$nome_a%' AND COGNOME LIKE '%$cognome_a%'
AND PAESE_NASCITA = $paese
```

$$\begin{aligned}
 AUTORI_RICHIESTI & \leftarrow \sigma_{<NOME=\%nome_a\% \wedge COGNOME=\%cognome_a\% \wedge PAESE_NASCITA=paese>} (Autore) \\
 OUT & \leftarrow \pi_{<NOME, COGNOME, ID_AUTORE, DATA_NASCITA, PAESE_NASCITA>} (AUTORI_RICHIESTI)
 \end{aligned}$$

[4]

Consultare l'elenco degli utenti della biblioteca (con le informazioni principali).

```
SELECT NOME, COGNOME, MATRICOLA, NUMERO_TELEFONO
FROM Studente
```

$$\pi_{<NOME, COGNOME, MATRICOLA, NUMERO_TELEFONO>} (Studente)$$

[5]

Ricerca di un utente della biblioteca e il suo storico dei prestiti (compresi quelli in corso).

```
SELECT p.ID_PRESTITO, p.DATA_USCITA, s.COGNOME, l.ISBN, l.ID_LIBRO
FROM Prestito AS p, Studente AS s, Libro AS l
WHERE MATRICOLA = '$matricola' AND p.MATRICOLA_S = s.MATRICOLA
AND p.ID_L = l.ID_LIBRO
```

$$\begin{aligned}
 & \rho_p(Prestito), \rho_s(Studente), \rho_l(Libro) \\
 JOIN_PSL & \leftarrow (Prestito) \bowtie_{p.MATRICOLA=s.MATRICOLA \wedge s.MATRICOLA=matricola} (Studente) \bowtie_{p.ID_L=l.ID_LIBRO} (Libro) \\
 OUT & \leftarrow \pi_{<p.ID_PRESTITO, p.DATA_USCITA, s.COGNOME, l.ISBN, l.ID_LIBRO>} (JOIN_PSL)
 \end{aligned}$$

[6]

Consultare lo storico dei prestiti comprese le informazioni (sintetiche - nome, cognome) sull'utente.

```
SELECT p.ID_PRESTITO, p.MATRICOLA_S, p.DATA_USCITA, s.NOME, s.COGNOME
FROM Prestito AS p, Studente AS s
WHERE p.MATRICOLA_S = s.MATRICOLA
```

$$\begin{aligned}
 & \rho_p(Prestito), \rho_s(Studente) \\
 PRESTITI_UTENTE & \leftarrow (Prestito) \bowtie_{<p.MATRICOLA=s.MATRICOLA>} (Studente) \\
 OUT & \leftarrow \pi_{<p.ID_PRESTITO, p.MATRICOLA_S, p.DATA_USCITA, s.NOME, s.COGNOME>} (PRESTITI_UTENTE)
 \end{aligned}$$

[7]

Ricerca dei prestiti effettuati in un range di date – nel caso in cui non vengano inserite date deve mostrare i prossimi in scadenza (quelli che scadranno in futuro) comprese le informazioni sintetiche sull'autore.

```
SELECT ID_PRESTITO, MATRICOLA_S, DATA_USCITA FROM Prestito
WHERE DATA_USCITA >= '$data_inizio'
AND DATA_USCITA <= '$data_fine'
```

$$IN_RANGE \leftarrow \sigma_{DATA_USCITA \geq data_inizio \wedge DATA_USCITA \leq data_fine} (Prestito)$$

$$OUT \leftarrow \pi_{ID_PRESTITO, MATRICOLA_S, DATA_USCITA} (IN_RANGE)$$

Vista la necessità di istruzioni condizionali per la costruzione della query che soddisfi la richiesta, mostriamo solo il caso base.

[8]

Statistiche (qui abbiamo deciso di utilizzare le join (per completezza))

[8.a] Numero di libri pubblicati in un determinato anno.

```
SELECT i.ANNO_PUBBLICAZIONE AS anno, COUNT(i.ISBN) AS numero_libri
FROM ISBN_Info AS i
GROUP BY i.ANNO_PUBBLICAZIONE
ORDER BY i.ANNO_PUBBLICAZIONE
```

$$OUT \leftarrow ANNO_PUBBLICAZIONE \mathcal{F} COUNT_{ISBN}(\pi_{ANNO_PUBBLICAZIONE, ISBN}(ISBN_Info))$$

[8.b] Numero di prestiti effettuati in una determinata succursale.

```
SELECT s.NOME AS nome_succ, COUNT(p.ID_PRESTITO) AS numero_prestiti
FROM Succursale s
LEFT JOIN Libro l ON s.ID_SUCC = l.ID_S
LEFT JOIN Prestito p ON l.ID_LIBRO = p.ID_L
GROUP BY s.ID_SUCC
```

$$JOIN_LSP \leftarrow (Succursale) \bowtie_{LEFT < s.ID_SUCC=l.ID_S >} (Libro) \bowtie_{LEFT < l.ID_LIBRO=p.ID_L >} (Prestito)$$

$$OUT \leftarrow s.ID_SUCC \mathcal{F} COUNT_{p.ID_PRESTITO} (JOIN_LSP)$$

[8.c] Numero di libri pubblicati per autore.

```
SELECT a.ID_AUTORE, a.NOME AS nome, a.COGNOME AS cognome, COUNT(L.ID_LIBRO) AS numero_libri
FROM Autore a
JOIN Scritto_Da sd ON a.ID_AUTORE = sd.ID_A
JOIN Libro l ON SD.ID_L = l.ID_LIBRO
GROUP BY a.ID_AUTORE
```

$$\rho_a(Autore), \rho_{sd}(Scritto_Da), \rho_l(Libro)$$

$$JOIN_ASL \leftarrow (Autore) \bowtie_{< a.ID_AUTORE=sd.ID_A >} (Scritto_Da) \bowtie_{< l.ID_L=p.ID_LIBRO >} (Libro)$$

$$OUT \leftarrow ID_AUTORE \mathcal{F} COUNT_{l.ID_L} (JOIN_ASL)$$