

RELAZIONE PROGETTO BASI DI DATI ANNO 2022/2023
GRUPPO N. 6, GAIA MARZOLA – SOLOMON OLAMIDE TAIWO

La biblioteca dell'Università degli Studi di Ferrara ha richiesto la realizzazione di una base di dati per gestire le informazioni sui libri, gli utenti e i prestiti effettuati. L'obiettivo principale è fornire una piattaforma web per consentire agli utenti di consultare il catalogo della biblioteca e accedere alle informazioni relative ai prestiti. Per procedere alla realizzazione di questo progetto, si è iniziato definendo un **diagramma ER** appositamente studiato, che segue le direttive della traccia offerta: le entità del database sono:

- **DIPARTIMENTO**: "CodDip" è la chiave primaria, "Nome" e "Indirizzo" sono attributi (il secondo è un attributo composto che a sua volta è costituito dagli attributi "Città", "Cap", "Via", "NCivico");
- **LIBRO**: avente "CodLibro" come chiave primaria, gli attributi di quest'entità sono "AnnoPubb", "Titolo", "ISBN" e "Lingua";
- **AUTORE**: la chiave primaria è "CodAutore", mentre "LuogoNascita", "Nome", "Cognome" e "DataNascita" sono gli attributi;
- **UTENTE**: la chiave primaria di quest'ultima entità è "NMatricola", mentre "Cognome", "Nome" e "NTelefono" sono attributi semplici. Abbiamo infine "Indirizzo", che è un attributo composto – come nell'entità "Dipartimento" da "Città", "Cap", "Via", "NCivico".

Tra le entità vi sono tre relazioni differenti:

- **CUSTODIRE**: relazione uno a molti, un libro deve essere custodito da uno e un solo dipartimento ed un dipartimento deve custodire uno o più libri;
- **SCRIVERE**: relazione molti a molti, un libro deve essere scritto da uno o più autori e l'autore deve scrivere uno o più libri (per definirsi tale);
- **PRESTITO**: relazione molti a molti, un utente può prendere in prestito nessuno o più libri e un libro può essere preso in prestito da nessuno o più utenti.

Di seguito viene definito lo **schema logico** (con le chiavi esterne in grassetto e sottolineate) ed il **modello relazionale in terza forma normale (3FN)** con i relativi vincoli:

Libro (CodLibro, Titolo, ISBN, Lingua, AnnoPubb, **CodDip**)
Autore (**CodAutore**, Nome, Cognome, DataNascita, **LuogoNascita**)
Utente (**NMatricola**, Nome, Cognome, Via, NCivico, Cap, Città, **NTelefono**)
Dipartimento (**CodDip**, Nome, Via, NCivico, Cap, Città)
Scrivere (**Codice**, **CodLibro**, **CodAutore**)
Prestito (**CodPrestito**, Restituzione, DataUscita, **CodLibro**, **NMatricola**)

Per implementare le interrogazioni richieste dalla traccia del progetto, sono state utilizzate varie espressioni **SQL**, di seguito riportate con l'equivalente espressione scritta in **algebra relazionale**.

Query 1

```
SELECT Libro.CodLibro, Libro.Titolo, Libro.ISBN, Libro.Lingua, Libro.AnnoPubb, Libro.CodDip
FROM BibliotecaUNIFE.Libro
WHERE Titolo LIKE '%' . $nomeLibro . '%';
```

π CodLibro, Titolo, ISBN, Lingua, AnnoPubb, CodDip (σ Titolo LIKE '%nomeLibro%', Libro)

Query 2

```
SELECT Autore.CodAutore, Autore.Nome, Autore.Cognome, Autore.DataNascita, Autore.LuogoNascita
FROM BibliotecaUNIFE.Autore";
```

π CodAutore, Nome, Cognome, DataNascita, LuogoNascita (Autore)

Query 3

```
SELECT Autore.CodAutore, Autore.Nome, Autore.Cognome, Autore.DataNascita, Autore.LuogoNascita
FROM BibliotecaUNIFE.Autore
```

WHERE Autore.Nome LIKE '%' . \$nome . '%' **AND** Autore.Cognome LIKE '%' . \$cognome . '%' **AND** Autore.DataNascita LIKE '%' . \$data . '%' **AND** Autore.LuogoNascita LIKE '%' . \$luogo . '%';

σ Nome LIKE '%nome%' **AND** Cognome LIKE '%cognome%' **AND** DataNascita LIKE '%data%' **AND** LuogoNascita LIKE '%luogo%' (π CodAutore, Nome, Cognome, DataNascita, LuogoNascita (Autore))

Query 4

SELECT *
FROM BibliotecaUNIFE.Utente;

π * (Utente)

Query 5

SELECT *
FROM BibliotecaUNIFE.Utente
WHERE Utente.Nome LIKE '%' . \$nome . '%' **AND** Utente.Cognome LIKE '%' . \$cognome . '%';

σ Nome LIKE '%nome%' **AND** Cognome LIKE '%cognome%' (π * (Utente))

SELECT *
FROM BibliotecaUNIFE.Utente
WHERE Utente.NMatricola = '\$matricola' **AND** Utente.Nome LIKE '%' . \$nome . '%' **AND** Utente.Cognome LIKE '%' . \$cognome . '%';

σ NMatricola = 'matricola' **AND** Nome LIKE '%nome%' **AND** Cognome LIKE '%cognome%' (π * (Utente))

Query 6

SELECT Utente.NMatricola, Utente.Nome, Utente.Cognome, Libro.Titolo, Libro.ISBN,
Prestito.DataUscita, Prestito.Restituzione, Dipartimento.Nome, Dipartimento.Via, Dipartimento.NCivico,
Dipartimento.Cap, Dipartimento.Città
FROM Utente, Prestito, Libro, Dipartimento
WHERE Utente.NMatricola = Prestito.NMatricola **AND** Libro.CodLibro = Prestito.CodLibro **AND** Libro.CodDip =
Dipartimento.CodDip
ORDER BY Prestito.DataUscita **DESC**;

ρ Ut(NMatricola, Nome, Cognome) (Utente) \bowtie (π CodLibro, DataUscita, Restituzione, NMatricola, CodDip (σ CodLibro =
Prestito.CodLibro (Libro \bowtie Prestito))))
 \bowtie (π CodDip, Nome, Via, NCivico, Cap, Città (Dipartimento))

Query 7

SELECT Utente.NMatricola, Utente.Nome, Utente.Cognome, Utente.NTelefono, Utente.Via, Utente.NCivico,
Utente.Cap, Utente.Città, Libro.Titolo, Libro.ISBN, Prestito.DataUscita, Prestito.Restituzione, Dipartimento.Nome,
Dipartimento.Via, Dipartimento.NCivico, Dipartimento.Cap, Dipartimento.Città
FROM Utente, Prestito, Libro, Dipartimento
WHERE Utente.NMatricola = Prestito.NMatricola **AND** Libro.CodLibro = Prestito.CodLibro **AND** Libro.CodDip =
Dipartimento.CodDip **AND** Prestito.Restituzione = 0 **AND** Prestito.DataUscita > date_sub(current_date(), INTERVAL
'\$giorni' DAY)
ORDER BY Prestito.DataUscita;

ρ Ut(NMatricola, Nome, Cognome, NTelefono, Via, NCivico, Cap, Città) (Utente) \bowtie (π CodLibro, DataUscita, Restituzione,
NMatricola, CodDip (σ Restituzione = 0 **AND** DataUscita > date_sub(current_date(), INTERVAL giorni DAY) (Libro \bowtie
Prestito)))) \bowtie (π CodDip, Nome, Via, NCivico, Cap, Città (Dipartimento))

SELECT Utente.NMatricola, Utente.Nome, Utente.Cognome, Utente.NTelefono, Utente.Via, Utente.NCivico,
Utente.Cap, Utente.Città, Libro.Titolo, Libro.ISBN, Prestito.DataUscita, Prestito.Restituzione, Dipartimento.Nome,
Dipartimento.Via, Dipartimento.NCivico, Dipartimento.Cap, Dipartimento.Città
FROM Utente, Prestito, Libro, Dipartimento
WHERE Utente.NMatricola = Prestito.NMatricola **AND** Libro.CodLibro = Prestito.CodLibro **AND** Libro.CodDip =
Dipartimento.CodDip **AND** Prestito.DataUscita **BETWEEN** '\$Data1' **AND** '\$Data2'
ORDER BY Prestito.DataUscita;

ρ Ut(NMatricola, Nome, Cognome, NTelefono, Via, NCivico, Cap, Città) (Utente) \bowtie (π CodLibro, DataUscita, Restituzione, NMatricola, CodDip (σ DataUscita **BETWEEN** 'Data1' **AND** 'Data2' (Libro \bowtie Prestito)))) \bowtie (π CodDip, Nome, Via, NCivico, Cap, Città (Dipartimento))

Query 8a

```
SELECT COUNT(Libro.CodLibro)
FROM Libro
WHERE Libro.AnnoPubb = '$Anno';
```

ρ CountLibro (COUNT(CodLibro)) (σ AnnoPubb = 'Anno' (Libro))

Query 8b

```
SELECT *
FROM BibliotecaUNIFE.Dipartimento;
```

π * (Dipartimento)

Query 8c

```
SELECT Autore.CodAutore, Autore.Nome, Autore.Cognome, Autore.DataNascita, Autore.LuogoNascita,
COUNT(Libro.CodLibro) AS N_libri
FROM Scrivere, Libro, Autore
WHERE Scrivere.CodAutore = Autore.CodAutore AND Scrivere.CodLibro = Libro.CodLibro
GROUP BY Autore.CodAutore;
```

ρ Temp1 (π CodAutore, CodLibro (Scrivere)) \bowtie Libro \bowtie Autore ρ N_libri (COUNT(CodLibro) AS N_libri) (π CodAutore, Nome, Cognome, DataNascita, LuogoNascita (Temp1))

Query 9

Query 10

Query potenzialmente interessante perché permette di capire quale libro viene prestato maggiormente, permettendo ai dipartimenti di eventualmente provvedere all'ottenimento di maggiori copie del suddetto.

```
SELECT Libro.CodLibro, Libro.Titolo, Dipartimento.Nome, Libro.Lingua, COUNT(Prestito.CodLibro) AS
NumeroPrestiti
FROM Libro
JOIN Prestito ON Libro.CodLibro = Prestito.CodLibro
JOIN Dipartimento ON Libro.CodDip = Dipartimento.CodDip
GROUP BY Libro.CodLibro, Libro.Titolo, Dipartimento.Nome, Libro.Lingua
ORDER BY NumeroPrestiti DESC
LIMIT 1;
```

ρ Temp1 (π CodLibro, Titolo, CodDip, Lingua (Libro)) \bowtie Prestito \bowtie Dipartimento

ρ NumeroPrestiti (COUNT(CodLibro) AS NumeroPrestiti) (π CodLibro, Titolo, Nome, Lingua (Temp1)) σ NumeroPrestiti = MAX(NumeroPrestiti) (NumeroPrestiti)