

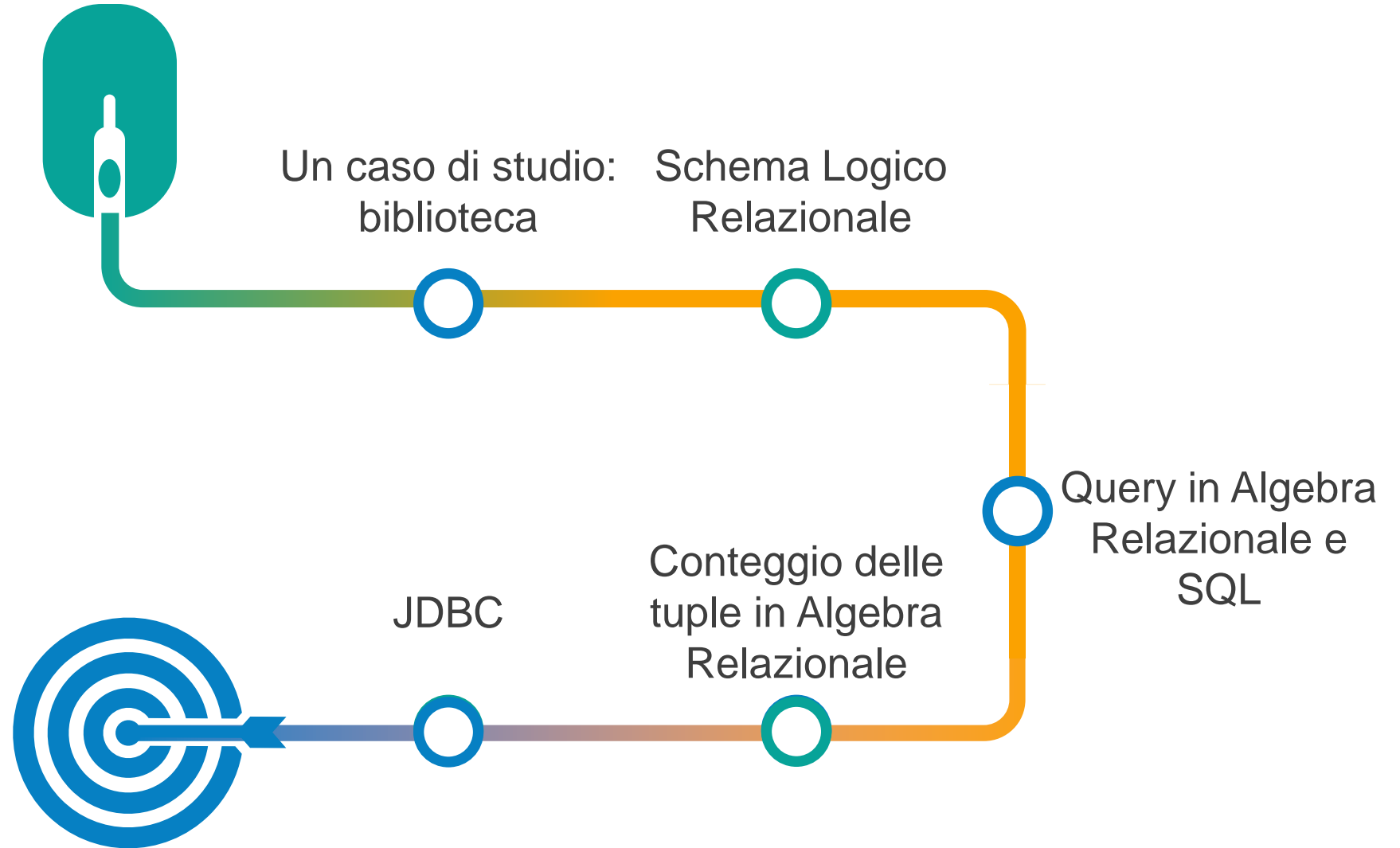


BASI DI DATI QUERY

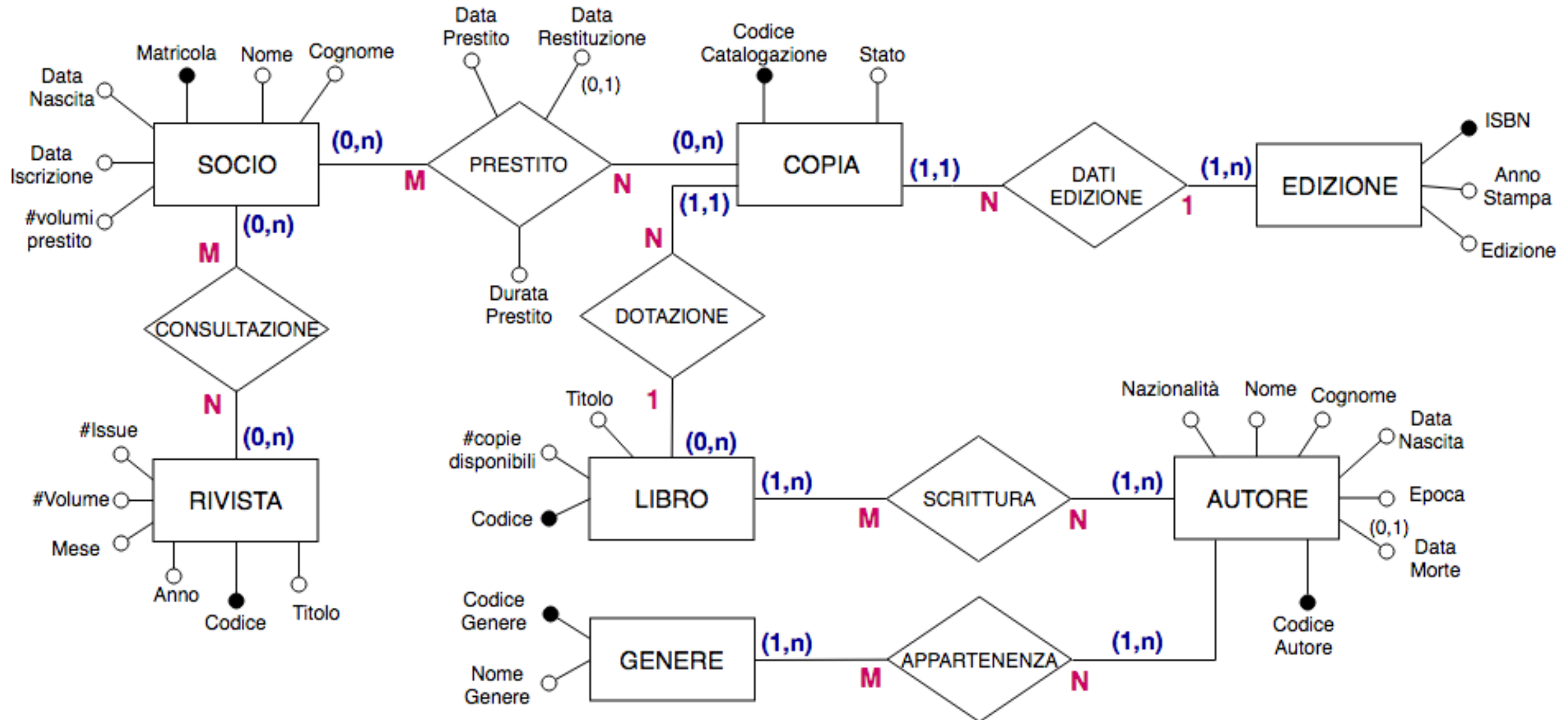
Polese G. Caruccio L. Breve B.

a.a. 2023/2024

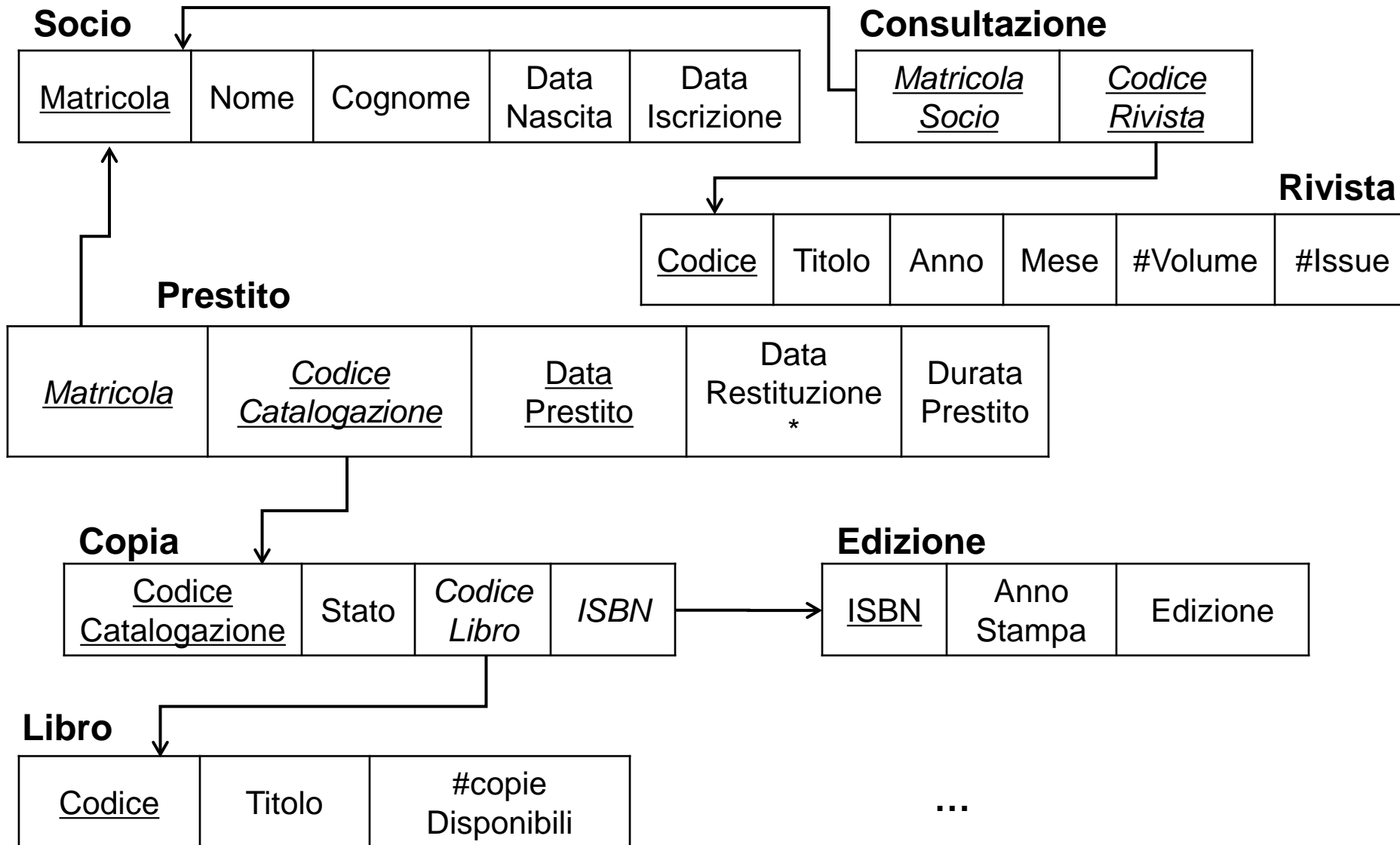
Outline



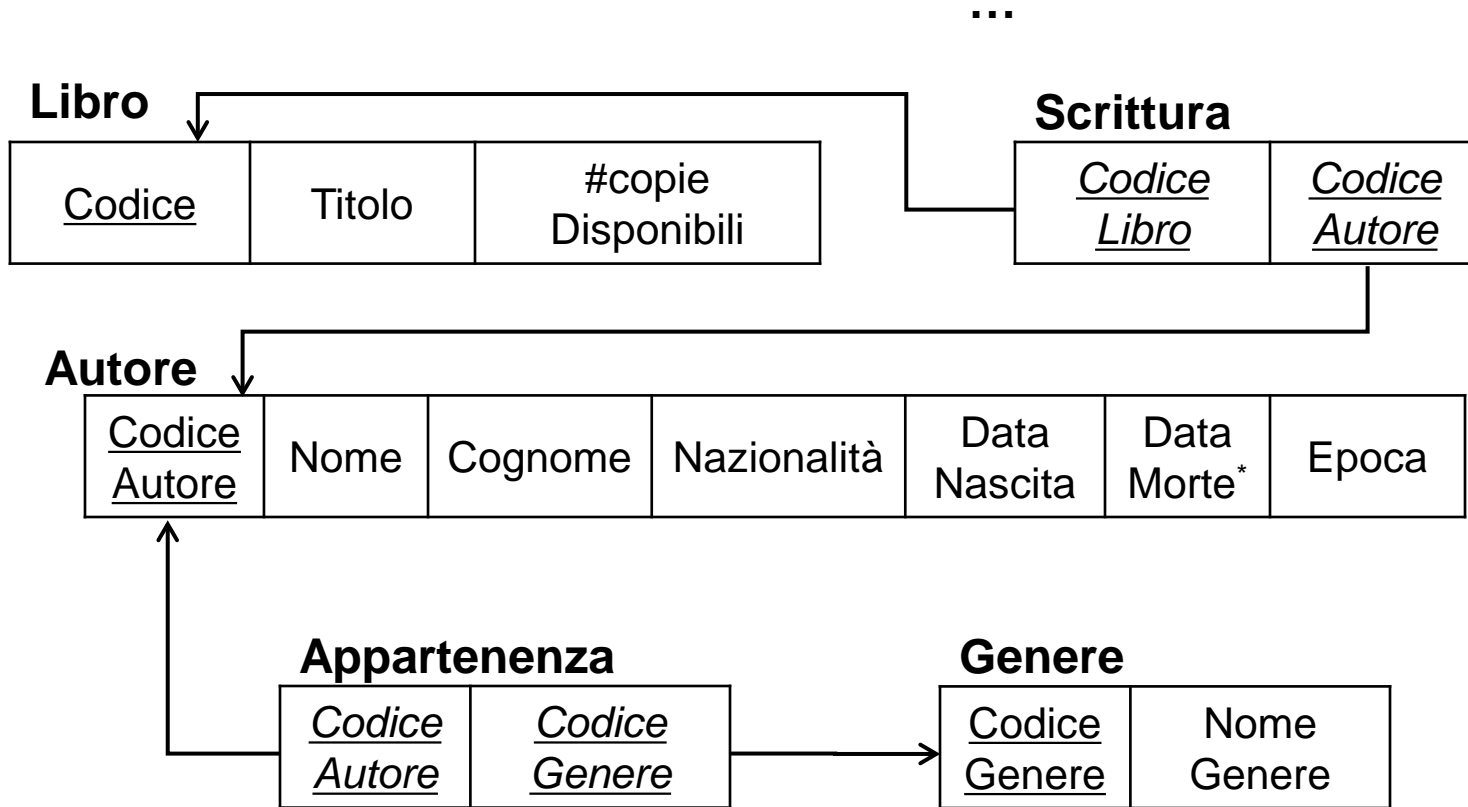
SCHEMA RISTRUTTURATO



MAPPING NELLO SCHEMA LOGICO RELAZIONALE



MAPPING NELLO SCHEMA LOGICO RELAZIONALE



MODIFICA ALLO SCHEMA (1)

- Il comando CHANGE COLUMN

```
ALTER TABLE table_name  
CHANGE COLUMN old_name new_name column_type;
```

- Esempio

- Ridenominare l'attributo numCopieDisponibili nella tabella libro, chiamandolo copieDisponibili

```
ALTER TABLE libro
```

```
CHANGE COLUMN numCopieDisponibili  
copieDisponibili INT NOT NULL;
```

MODIFICA ALLO SCHEMA (2)

- Il comando ADD COLUMN

```
ALTER TABLE table_name  
ADD COLUMN column_name column_type;
```

- Esempio

- Aggiungiamo l'attributo copieDisponibili nella tabella libro

```
ALTER TABLE libro  
ADD COLUMN copieDisponibili INT;
```

MODIFICA ALLO SCHEMA (3)

- Il comando DROP COLUMN

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

- Esempio

- Cancelliamo l'attributo copieDisponibili nella tabella libro

```
ALTER TABLE libro  
DROP COLUMN copieDisponibili;
```


MODIFICA ALLO SCHEMA (4)

- Eliminare **NOT NULL** sull'attributo ISBN di copia;

```
ALTER TABLE copia  
CHANGE COLUMN ISBN  
ISBN VARCHAR(16);
```

INSERIMENTO DEI DATI (1)

- Inseriamo i seguenti dati
 - Libro
 - Nome: I promessi sposi
 - CopieDisponibili: 3
 - Autore
 - Nome: Alessandro
 - Cognome: Manzoni
 - Nazionalità: Italiana
 - DataNascita: 1785-03-07
 - DataMorte: 1873-05-22
 - Epoca: Romanticismo
 - Scrittura
 - CodiceLibro: 1
 - CodiceAutore: 1

INSERIMENTO DEI DATI (2)

- Inseriamo i seguenti dati
 - Edizione
 - ISBN: 978-8844039011
 - AnnoStampa: 2010
 - Edizione: 2
 - Copia
 - Stato: nuovo
 - CodiceLibro: 1
 - ISBN: 978-8844039011
 - Copia (2 volte)
 - Stato: nuovo
 - CodiceLibro: 1

QUERY DI INTERROGAZIONE

- Proviamo le seguenti query

```
SELECT Nome, Cognome  
FROM socio INNER JOIN prestito ON  
    socio.matricola = prestito.matricola;
```

QUERY DI INTERROGAZIONE

- Proviamo le seguenti query

```
SELECT *  
FROM copia JOIN edizione ON  
      copia.ISBN = edizione.ISBN;
```

QUERY DI INTERROGAZIONE

- Proviamo le seguenti query

```
SELECT *  
FROM copia LEFT OUTER JOIN edizione ON  
      copia.ISBN = edizione.ISBN;
```

QUERY DI INTERROGAZIONE

- Proviamo le seguenti query

```
SELECT *  
FROM copia RIGHT OUTER JOIN edizione ON  
      copia.ISBN = edizione.ISBN;
```

QUERY DI INTERROGAZIONE

- Proviamo le seguenti query

```
SELECT *  
FROM copia CROSS JOIN edizione
```

- Questa corrisponde al prodotto cartesiano

```
SELECT *  
FROM copia,edizione
```


QUERY DI INTERROGAZIONE

- Proviamo le seguenti query

SELECT *

FROM copia NATURAL JOIN edizione

QUERY IN SQL: ESEMPIO 1

- Trovare le copie dei libri di cui non si conosce l'edizione (titolo libro, codice catalogazione)
- Algebra Relazionale e SQL: Elencare i soci che hanno consultato almeno una rivista, e che abbiano preso in prestito almeno un libro (cognome e nome)
- Algebra Relazionale e SQL: Selezionare gli autori (nome, cognome) che abbiano scritto in uno ed un solo genere

QUERY IN SQL: ESEMPIO 1

- Trovare le copie dei libri di cui non si conosce l'edizione (titolo libro, codice catalogazione)

```
SELECT codiceCatalogazione, titolo
FROM libro JOIN copia ON libro.codice = copia.codiceLibro
WHERE ISBN IS NULL;
```

QUERY IN ALGEBRA RELAZIONALE: ESEMPIO 2

- Elencare i clienti che hanno consultato almeno una rivista, e che abbiano preso in prestito almeno un libro (cognome e nome)

$$\text{PROJ}_{\text{Nome, Cognome}}(\text{Socio JOIN}_{\text{matricola=matricolaSocio}} \text{Consultazione})$$

$$\cap$$

$$\text{PROJ}_{\text{Nome, Cognome}}(\text{Socio JOIN} \text{Prestito})$$

QUERY IN SQL: ESEMPIO 2

- Elencare i soci che hanno consultato almeno una rivista, e che abbiano preso in prestito almeno un libro (cognome e nome)

```
SELECT cognome, nome
FROM socio JOIN consultazione
ON socio.matricola=consultazione.matricolaSocio
WHERE matricola IN (
    SELECT matricola
    FROM prestito);
```

QUERY IN ALGEBRA RELAZIONALE: ESEMPIO 3

- Selezionare gli autori (nome, cognome) che abbiamo scritto in uno ed un solo genere

PROJ_{Nome, Cognome}(Autore **JOIN**_{Codice=CodiceAutore} Appartenenza)

-

(**PROJ**_{Nome, Cognome}(Autore **JOIN**_{Codice=CodiceAutore}

(**SEL**_{CodiceGenere<>CodiceGenere1}

(Appartenenza

JOIN

(**REN**_{CodiceGenere1←CodiceGenere}(Appartenenza))))

QUERY IN SQL: ESEMPIO 3

- Selezionare gli autori che abbiamo scritto in uno ed un solo genere

```
SELECT codiceAutore, nome, cognome, COUNT(DISTINCT  
    codiceGenere) AS numGeneri  
FROM autore JOIN appartenenza  
    ON autore.codice = appartenenza.codiceAutore  
GROUP BY codiceAutore, nome, cognome  
HAVING numGeneri=1;
```

QUERY IN SQL: ESEMPIO 3

- Selezionare gli autori (nome e cognome) che abbiamo scritto in uno ed un solo genere

```
SELECT nome, cognome
FROM (SELECT codiceAutore, nome, cognome,
      COUNT(DISTINCT codiceGenere) AS numGeneri
      FROM autore JOIN appartenenza
      ON autore.codice=appartenenza.codiceAutore
      GROUP BY codiceAutore, nome, cognome
      HAVING numGeneri=1) autoreConUno;
```


QUERY IN SQL: ESEMPIO 3

- Selezionare gli autori (nome, cognome) che abbiamo scritto in uno ed un solo genere

```
SELECT nome, cognome
FROM autore
WHERE codice IN
  (SELECT codiceAutore
   FROM appartenenza
   GROUP BY codiceAutore
   HAVING COUNT(DISTINCT codiceGenere) =1);
```

QUERY IN SQL: ESEMPIO 3

- Selezionare gli autori (nome, cognome) che abbiamo scritto in uno ed un solo genere

```
SELECT nome, cognome
FROM autore JOIN appartenenza
      ON autore.codice=appartenenza.codiceAutore
GROUP BY codiceAutore, nome, cognome
HAVING COUNT(DISTINCT codiceGenere) = 1;
```

UN PROGRAMMA CON JDBC

```
import java.sql.*;
public class PrimoJDBC {
    public static void main(String[] arg){
        Connection con = null ;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            String url = "jdbc:mysql://localhost:3306/biblioteca"
                + "?useUnicode=true&useJDBCCompliantTimezoneShift=true"
                + "&useLegacyDatetimeCode=false&serverTimezone=UTC";
            String username = "<username>"; String pwd = "<pwd>";
            con = DriverManager.getConnection(url,username,pwd);
        }
        catch(Exception e){
            System.out.println("Connessione fallita");
        }
        try {
            Statement query = con.createStatement();
            ResultSet result =
                query.executeQuery("select * from socio");
            while (result.next()){
                String nomeSocio = result.getString("Nome");
                System.out.println(nomeSocio);
            }
        }
        catch (Exception e){
            System.out.println("Errore nell'interrogazione");
        }
    }
}
```