

JPA - PostgreSQL - Relazioni

1. Creazione progetto IntelliJ

- File > New > Project
- Seleziona **Java**
- Seleziona **Maven** come builder
- Crea progetto con struttura di base

2. Struttura delle cartelle

```
my-jpa-project/
├── pom.xml
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   ├── it/
│   │   │   │   ├── epicode/
│   │   │   │   │   ├── dao/
│   │   │   │   │   │   ├── AutoreDAO.java
│   │   │   │   │   │   ├── CasaEditriceDAO.java
│   │   │   │   │   │   ├── CategoriaDAO.java
│   │   │   │   │   │   └── LibroDAO.java
│   │   │   │   │   ├── model/
│   │   │   │   │   │   ├── Autore.java
│   │   │   │   │   │   ├── CasaEditrice.java
│   │   │   │   │   │   ├── Categoria.java
│   │   │   │   │   │   └── Libro.java
│   │   │   │   │   └── Main.java
│   │   └── resources/
│   │       ├── META-INF/
│   │       └── persistence.xml
```

Se la cartella `META-INF` non esiste, crearla manualmente in `resources`.

3. Creazione DB in pgAdmin

1. Apri **pgAdmin**
2. Tasto destro su *Databases* > **Create** > **Database...**
3. Inserisci il nome del DB (es. `epicode_db`)
4. Conferma

4. Configurazione persistence.xml

Percorso: src/main/resources/META-INF/persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.2"
    xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
    http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd">
  <persistence-unit name="epicode">
    <properties>
      <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://local
      <property name="javax.persistence.jdbc.user" value="postgres"/>
      <property name="javax.persistence.jdbc.password" value="XXXXXX"/>
      <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver
      <property name="hibernate.default_schema" value="public"/>
      <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQL
      <property name="hibernate.hbm2ddl.auto" value="create"/>
    </properties>
  </persistence-unit>
</persistence>
```

Sostituire:

- XXXXXX con la password di PostgreSQL
- YYYYYYYYYYYY con il nome del database (es. epicode_db)

Significato hibernate.hbm2ddl.auto

- create : crea ogni volta le tabelle da zero
- update : aggiorna tabelle senza cancellare
- validate : controlla solo che il mapping corrisponda
- none : nessuna azione

5. Annotazioni principali JPA

```
@Entity // indica che è una classe persistente
@Table(name = "nome_tabella") // definisce il nome tabella, in minuscolo
@Column(name = "colonna", length = 100, unique = true) // nome colonna, lunghezza mass
@Id // indica la chiave primaria
@GeneratedValue(strategy = GenerationType.IDENTITY) // auto-generazione dell'ID
@Enumerated(EnumType.STRING) // salva enum come stringa
@ManyToOne // relazione molti-a-uno
@OneToMany(mappedBy = "...") // uno-a-molti, 'mappedBy' specifica il campo dell'altro
```

```
@ManyToMany // relazione multi-a-molti
```

6. Pom.xml

```
<dependencies>
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>6.2.5.Final</version>
  </dependency>
  <dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>42.7.1</version>
  </dependency>
</dependencies>
```

7. Classi Entity e DAO

Le entità `Autore`, `CasaEditrice`, `Categoria` e `Libro` si trovano nella cartella `model`, e sono annotate correttamente con `@Entity`, `@Table`, `@Id`, `@Column`, `@ManyToOne`, `@OneToMany`, `@ManyToMany` a seconda delle relazioni. Includono:

- Costruttore vuoto
- Tutti i getter/setter
- `equals`, `hashCode`, `toString`

Per ciascuna, esiste un DAO con metodi `save`, `getById`, `delete`, con `EntityManager` passato nel costruttore.

8. Classi

Inizio con la prima entità: `Categoria`.

Categoria.java

```

package it.epicode.model;

import jakarta.persistence.*;
import java.util.Objects;

@Entity // Questa classe rappresenta una tabella nel database
@Table(name = "categoria") // Nome della tabella in minuscolo
public class Categoria {

    @Id // Identifica la chiave primaria
    @GeneratedValue(strategy = GenerationType.IDENTITY) // Auto-generazione dell'ID
    private Long id;

    @Column(name = "nome", length = 100, unique = true) // Campo 'nome' con massimo 10
    private String nome;

    public Categoria() {}

    public Categoria(String nome) {
        this.nome = nome;
    }

    public Long getId() {
        return id;
    }

    public String getNome() {
        return nome;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Categoria)) return false;
        Categoria that = (Categoria) o;
        return Objects.equals(id, that.id);
    }

    @Override
    public int hashCode() {
        return Objects.hash(id);
    }

    @Override

```

```

        public String toString() {
            return "Categoria{" +
                "id=" + id +
                ", nome='" + nome + '\'' +
                '}';
        }
    }

```

CategoriaDAO.java

```

package it.epicode.dao;

import it.epicode.model.Categoria;
import jakarta.persistence.EntityManager;

public class CategoriaDAO {
    private EntityManager em;

    public CategoriaDAO(EntityManager em) {
        this.em = em;
    }

    public void save(Categoria categoria) {
        em.persist(categoria);
    }

    public Categoria getById(Long id) {
        return em.find(Categoria.class, id);
    }

    public void delete(Categoria categoria) {
        if (!em.contains(categoria)) {
            categoria = em.merge(categoria);
        }
        em.remove(categoria);
    }
}

```

Autore.java

```

package it.epicode.model;

import jakarta.persistence.*;
import java.util.Objects;

@Entity // Rappresenta una tabella nel database
@Table(name = "autore") // Nome tabella minuscolo

```

```
public class Autore {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "nome", length = 100)
    private String nome;

    @Column(name = "cognome", length = 100)
    private String cognome;

    public Autore() {}

    public Autore(String nome, String cognome) {
        this.nome = nome;
        this.cognome = cognome;
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getCognome() {
        return cognome;
    }

    public void setCognome(String cognome) {
        this.cognome = cognome;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Autore)) return false;
        Autore autore = (Autore) o;
        return Objects.equals(id, autore.id);
    }

    @Override
    public int hashCode() {
```

```

        return Objects.hash(id);
    }

    @Override
    public String toString() {
        return "Autore{" +
            "id=" + id +
            ", nome='" + nome + '\'' +
            ", cognome='" + cognome + '\'' +
            '}';
    }
}

```

AutoreDAO.java

```

package it.epicode.dao;

import it.epicode.model.Autore;
import jakarta.persistence.EntityManager;

public class AutoreDAO {
    private EntityManager em;

    public AutoreDAO(EntityManager em) {
        this.em = em;
    }

    public void save(Autore autore) {
        em.persist(autore);
    }

    public Autore getById(Long id) {
        return em.find(Autore.class, id);
    }

    public void delete(Autore autore) {
        if (!em.contains(autore)) {
            autore = em.merge(autore);
        }
        em.remove(autore);
    }
}

```

CasaEditrice.java

```

package it.epicode.model;

```

```

import jakarta.persistence.*;
import java.util.Objects;

@Entity
@Table(name = "casaeditrice") // nome tabella minuscolo
public class CasaEditrice {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "nome", length = 100, unique = true)
    private String nome;

    public CasaEditrice() {}

    public CasaEditrice(String nome) {
        this.nome = nome;
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof CasaEditrice)) return false;
        CasaEditrice that = (CasaEditrice) o;
        return Objects.equals(id, that.id);
    }

    @Override
    public int hashCode() {
        return Objects.hash(id);
    }

    @Override
    public String toString() {
        return "CasaEditrice{" +
            "id=" + id +

```



```

        ", nome='" + nome + '\'" +
        "'}";
    }
}

```

CasaEditriceDAO.java

```

package it.epicode.dao;

import it.epicode.model.CasaEditrice;
import jakarta.persistence.EntityManager;

public class CasaEditriceDAO {
    private EntityManager em;

    public CasaEditriceDAO(EntityManager em) {
        this.em = em;
    }

    public void save(CasaEditrice casaEditrice) {
        em.persist(casaEditrice);
    }

    public CasaEditrice getById(Long id) {
        return em.find(CasaEditrice.class, id);
    }

    public void delete(CasaEditrice casaEditrice) {
        if (!em.contains(casaEditrice)) {
            casaEditrice = em.merge(casaEditrice);
        }
        em.remove(casaEditrice);
    }
}

```

... (contenuto precedente invariato) ...

Libro.java

```

package it.epicode.model;

import jakarta.persistence.*;
import java.util.List;
import java.util.Objects;

@Entity
@Table(name = "libro")

```

```

public class Libro {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "titolo", length = 200)
    private String titolo;

    @ManyToOne
    @JoinColumn(name = "autore_id")
    private Autore autore;

    @ManyToOne
    @JoinColumn(name = "casa_editrice_id")
    private CasaEditrice casaEditrice;

    @ManyToMany
    @JoinTable(
        name = "libro_categoria",
        joinColumns = @JoinColumn(name = "libro_id"),
        inverseJoinColumns = @JoinColumn(name = "categoria_id")
    )
    private List<Categoria> categorie;

    public Libro() {}

    public Libro(String titolo, Autore autore, CasaEditrice casaEditrice, List<Categoria> categorie) {
        this.titolo = titolo;
        this.autore = autore;
        this.casaEditrice = casaEditrice;
        this.categorie = categorie;
    }

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getTitolo() {
        return titolo;
    }

    public void setTitolo(String titolo) {
        this.titolo = titolo;
    }

    public Autore getAutore() {
        return autore;
    }
}

```

```

public void setAutore(Autore autore) {
    this.autore = autore;
}

public CasaEditrice getCasaEditrice() {
    return casaEditrice;
}

public void setCasaEditrice(CasaEditrice casaEditrice) {
    this.casaEditrice = casaEditrice;
}

public List<Categoria> getCategorie() {
    return categorie;
}

public void setCategorie(List<Categoria> categorie) {
    this.categorie = categorie;
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (!(o instanceof Libro)) return false;
    Libro libro = (Libro) o;
    return Objects.equals(id, libro.id);
}

@Override
public int hashCode() {
    return Objects.hash(id);
}

@Override
public String toString() {
    return "Libro{" +
        "id=" + id +
        ", titolo='" + titolo + '\'' +
        ", autore=" + autore +
        ", casaEditrice=" + casaEditrice +
        ", categorie=" + categorie +
        '}';
}

```

LibroDAO.java

```

package it.epicode.dao;

import it.epicode.model.Libro;

```

```

import jakarta.persistence.EntityManager;

public class LibroDAO {
    private EntityManager em;

    public LibroDAO(EntityManager em) {
        this.em = em;
    }

    public void save(Libro libro) {
        em.persist(libro);
    }

    public Libro getById(Long id) {
        return em.find(Libro.class, id);
    }

    public void delete(Libro libro) {
        if (!em.contains(libro)) {
            libro = em.merge(libro);
        }
        em.remove(libro);
    }
}

```

Ecco il codice completo della classe Main.java :

Main.java

```

package it.epicode;

import it.epicode.dao.*;
import it.epicode.model.*;
import jakarta.persistence.EntityManager;
import jakarta.persistence.EntityManagerFactory;
import jakarta.persistence.Persistence;

import java.util.List;

public class Main {
    public static void main(String[] args) {

        EntityManagerFactory emf = Persistence.createEntityManagerFactory("epicode");
        EntityManager em = emf.createEntityManager();

        CategoriaDAO categoriaDAO = new CategoriaDAO(em);
        AutoreDAO autoreDAO = new AutoreDAO(em);
        CasaEditriceDAO casaEditriceDAO = new CasaEditriceDAO(em);
        LibroDAO libroDAO = new LibroDAO(em);

        em.getTransaction().begin();
    }
}

```

```
// Inserimento 5 categorie
Categoria c1 = new Categoria("Fantasy");
Categoria c2 = new Categoria("Giallo");
Categoria c3 = new Categoria("Storia");
Categoria c4 = new Categoria("Horror");
Categoria c5 = new Categoria("Fantascienza");
categoriaDAO.save(c1);
categoriaDAO.save(c2);
categoriaDAO.save(c3);
categoriaDAO.save(c4);
categoriaDAO.save(c5);

// Inserimento 5 autori
Autore a1 = new Autore("Mario", "Rossi");
Autore a2 = new Autore("Luca", "Bianchi");
Autore a3 = new Autore("Anna", "Verdi");
Autore a4 = new Autore("Laura", "Neri");
Autore a5 = new Autore("Giulia", "Galli");
autoreDAO.save(a1);
autoreDAO.save(a2);
autoreDAO.save(a3);
autoreDAO.save(a4);
autoreDAO.save(a5);

// Inserimento 5 case editrici
CasaEditrice ce1 = new CasaEditrice("Mondadori");
CasaEditrice ce2 = new CasaEditrice("Einaudi");
CasaEditrice ce3 = new CasaEditrice("Feltrinelli");
CasaEditrice ce4 = new CasaEditrice("Rizzoli");
CasaEditrice ce5 = new CasaEditrice("Garzanti");
casaEditriceDAO.save(ce1);
casaEditriceDAO.save(ce2);
casaEditriceDAO.save(ce3);
casaEditriceDAO.save(ce4);
casaEditriceDAO.save(ce5);

// Inserimento 5 libri
Libro l1 = new Libro("Il primo libro", a1, ce1, List.of(c1, c2));
Libro l2 = new Libro("Avventure lontane", a2, ce2, List.of(c3));
Libro l3 = new Libro("La paura", a3, ce3, List.of(c4));
Libro l4 = new Libro("Il futuro oscuro", a4, ce4, List.of(c5));
Libro l5 = new Libro("Storie miste", a5, ce5, List.of(c1, c5));
libroDAO.save(l1);
libroDAO.save(l2);
libroDAO.save(l3);
libroDAO.save(l4);
libroDAO.save(l5);

// getById
System.out.println(categoriaDAO.getById(c1.getId()));
System.out.println(autoreDAO.getById(a1.getId()));
System.out.println(casaEditriceDAO.getById(ce1.getId()));
```

```
System.out.println(libroDAO.getById(l1.getId()));
```

```
// delete esempio  
libroDAO.delete(l5);
```

```
em.getTransaction().commit();  
em.close();
```

```
emf.close(); // importante per liberare le risorse del DB
```

```
}
```