

HIBERNATE

Tipi di fetch



Federico Parezzan



federico.parezzan@outlook.it



federico-parezzan



parez93

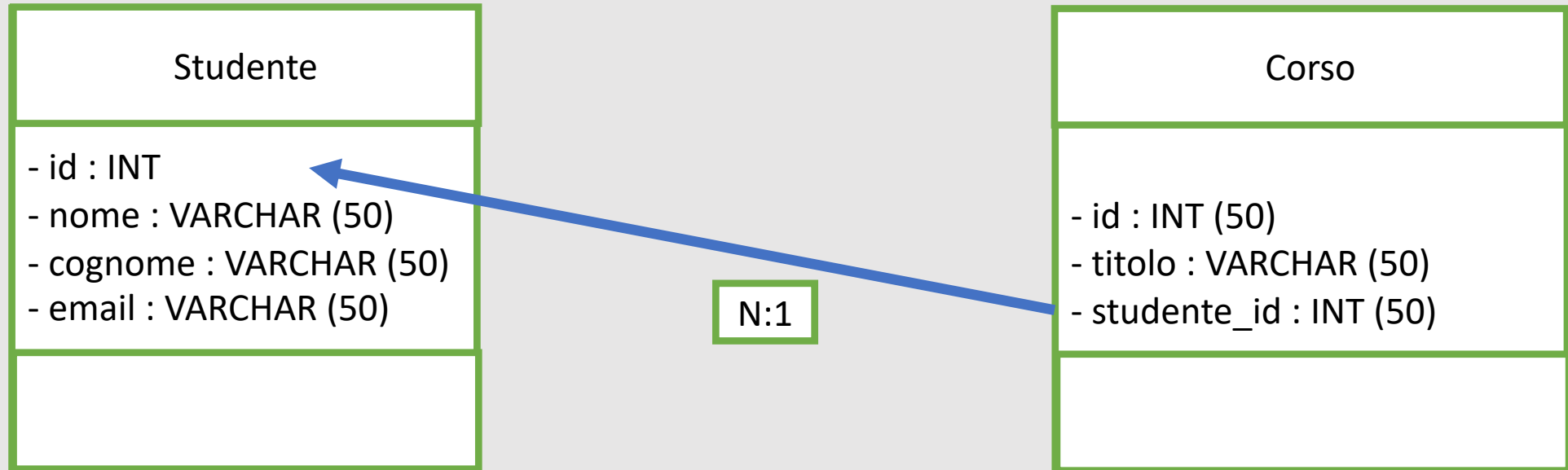
Agenda

- Fetch overview
- Fetch annotazioni
- Lazy fetch
- Recupero dei dati su richiesta

Fetch

1. **Eager** : viene recuperato tutto (anche le entità con vincolo referenziale)
2. **Lazy** : viene recuperato su richiesta (carica per prima solo l'entità principale)

Overview



Se recupero inizialmente solo il Corso e poi gli Studenti su richiesta → Lazy

Se recupero il Corso e gli Studenti associati → Eager

Best practice

- Recupera i dati quando ne hai assolutamente bisogno
- Il recupero Lazy è da preferire a quello Eager

Annotazioni

```
@OneToMany
@JoinColumn(fetch = FetchType.LAZY, mappedBy = "studente_id")
private List<Corso> corsi;
```

```
@OneToMany
@JoinColumn(fetch = FetchType.EAGER, mappedBy = "studente_id")
private List<Corso> corsi;
```

Default Annotazioni

Mapping	Default
OneToOne	Eager
OneToMany	Lazy
ManyToOne	Eager
ManyToMany	Lazy

Fetch Lazy

- Recupera i dati su richiesta
- Richiede una sessione HB aperta
- Se la sessione è chiusa, HB lancia un'eccezione (LazyInitializationException)

LazyInitializationException

```
session.beginTransaction();  
Studente s = session.get(Studente.class, id);  
session.getTransaction().commit();  
session.close();  
s.getCorsi()
```

- s.getCorsi() : lancia eccezione perché la sessione è chiusa e prova a fare una SELECT

Recupero dei dati su richiesta

- SELECT + getter appropriato dentro la sessione
- HQL

Recupero dei dati su richiesta

- SELECT + getter appropriato dentro la sessione
- HQL

Recupero tramite GET

```
session.beginTransaction();  
Studiante s = session.get(Studiante.class, id);  
s.getCorsi()  
session.getTransaction().commit();  
session.close();  
s.getCorsi()
```

- s.getCorsi() : nel primo caso HB va un'ulteriore select dopo il metodo GET
- s.getCorsi() : nel secondo caso il dato è già stato recuperato quindi non lancia l'eccezione

Recupero tramite HQL

```
Query<Studente> query = session.createQuery("SELECT i from Studente i"
                                             + " JOIN FETCH i.corso"
                                             + " WHERE i.id = :corsoId",
                                             Studente.class);
query.setParameter("corsoId", id);
Studente s = query.getSingleResult();
session.getTransaction().commit();
session.close();
s.getCorsi()
```

Q&A

DEMO