

# HIBERNATE

OneToOne



Federico Parezzan



federico.parezzan@outlook.it



federico-parezzan



parez93

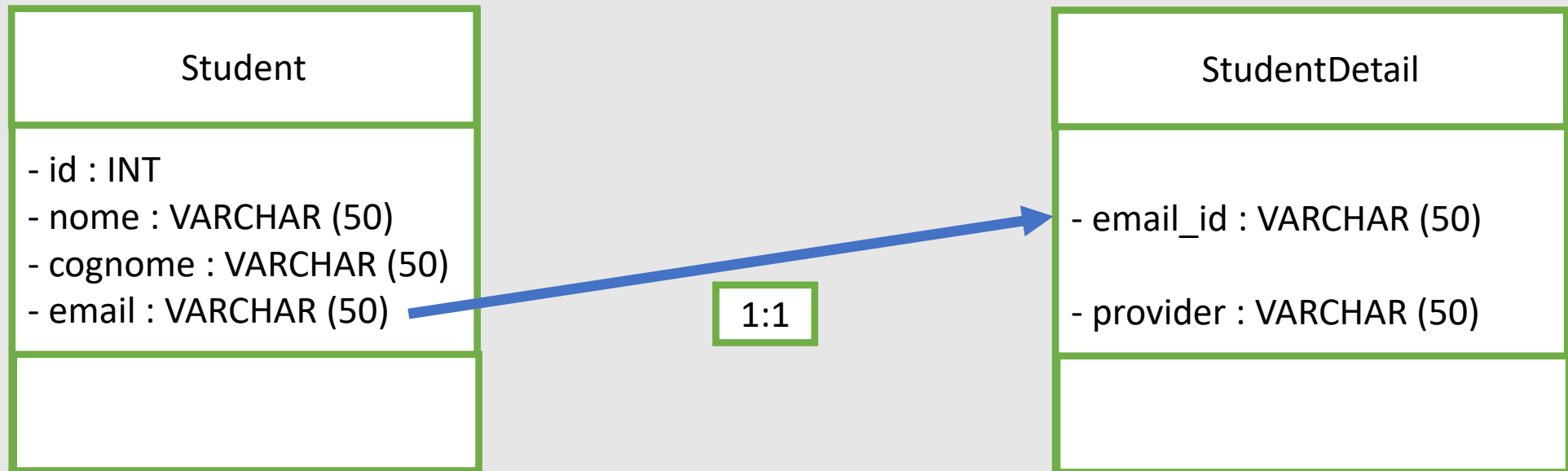
# Agenda

- OneToOne overview
- OneToOne
- OneToOne bi-direzionale
- Delete dettaglio

# Step

1. Definire in db la tabella con FK
2. Creare le entità che mappano le tabelle
3. Annotazione

# Overview



# Annotazione

```
@OneToOne  
@JoinColumn(name = "dettaglio_studente_id")  
private DettaglioStudente dettaglioStudente;
```

- @OneToOne : relazione 1:1
- @JoinColumn : mette in relazione le colonne di due entità
- dettaglio\_studente\_id : PK della tabella a cui la FK si riferisce
- DettaglioStudente dettaglioStudente : entità coinvolta nella relazione

# Esempio insert

```
Student s = new Student("Federico", "Parezzan");  
StudentDetail ds = new StudentDetail("mail@mail.com", "Gmail");  
s.setStudentDetail(ds);  
session.beginTransaction();  
session.save();  
session.getTransaction().commit();
```

- Associa in memoria i 2 oggetti (Student e StudentDetail)
- save() : salva su entrambe le tabelle se la politica è Cascade

# Bi-direzionale

Dalla classe StudentDetail voglio recuperare Student:

1. Aggiungere nuovo campo referente a Student
  2. Aggiungere getter/setter
  3. Aggiungere @OneToOne
- Nessun cambiamento a livello di database!

# Bi-direzionale annotazione

In DettaglioStudente:

```
@OneToOne(mappedBy="dettaglioStudente")  
private Student student;
```

In Studente:

```
@OneToOne  
@JoinColumn(name = "dettaglio_studente_id")  
private DettaglioStudente dettaglioStudente;
```



# mappedBy

```
@OneToOne(mappedBy="dettaglioStudente")  
private Student student;
```

- Guarda al campo *dettaglioStudente* nella classe Studente
- Usa l'informazione *@JoinColumn* nella classe Studente
- Aiuta a trovare lo studente associato

# Delete solo della classe dettaglio

Nella classe DettaglioStudente:

```
@OneToOne(mappedBy="dettaglioStudente",  
cascade = {CascadeType.DETACH, CascadeType.MERGE, CascadeType.REFRESH})  
private Student student;
```

- Importante non inserire CascadeType.REMOVE

# Delete solo della classe dettaglio

Nel classe contenente il main:

```
// ... select ...  
dettaglioStudiante.getStudiante().setDettaglioStudiante(null);  
session.delete(dettaglioStudiante);
```

- Basta rimuovere la referenza all'oggetto associato con NULL

# CascadeType - accenni

Detach	L'entità non è associata con una sessione di Hibernate
Merge	Se l'istanza è detach dalla sessione, il merge la riassocia alla sessione
Remove	Prende un'entità gestita e la rimuove
Refresh	Ricarica/sincronizza l'oggetto con il database

Q&A

DEMO