HIBERNATE

Configurazione tramite annotazioni

 Federico Parezzan

 federico.parezzan@outlook.it

 federico-parezzan

 parez93

Agenda

- Creare un file di configurazione di hibernate
- Annotare una classe Java
- Codice Java per operazioni su database

File di configurazione

- È un file che dice ad Hibernate come connettersi al database
- hibernate.cfg.xml
- La session_factory permette di ottenere gli oggetti di sessione per la connessione a database

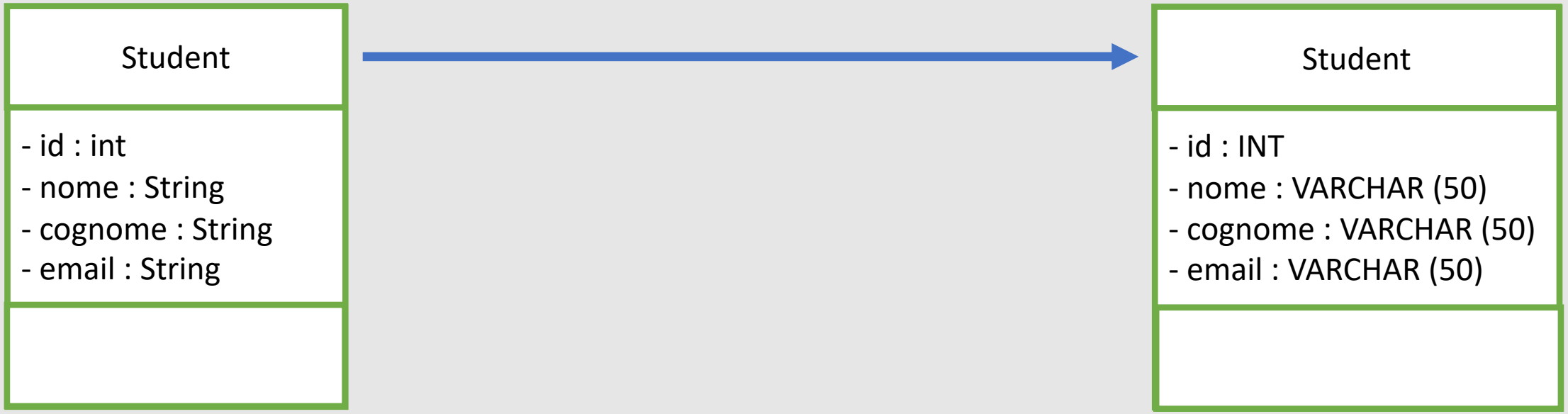
File di configurazione

```
<hibernate-configuration>
  <session-factory>
    <property
name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
    <property name="connection.url">
       jdbc:mysql://localhost:3306/hb_student_tracker?useSSL=false&amp;serv
erTimezone=UTC
    </property>
    <property name="connection.username">student</property>
    <property name="connection.password">student</property>
     ...
  </session-factory>
</hibernate-configuration>
```

DEMO

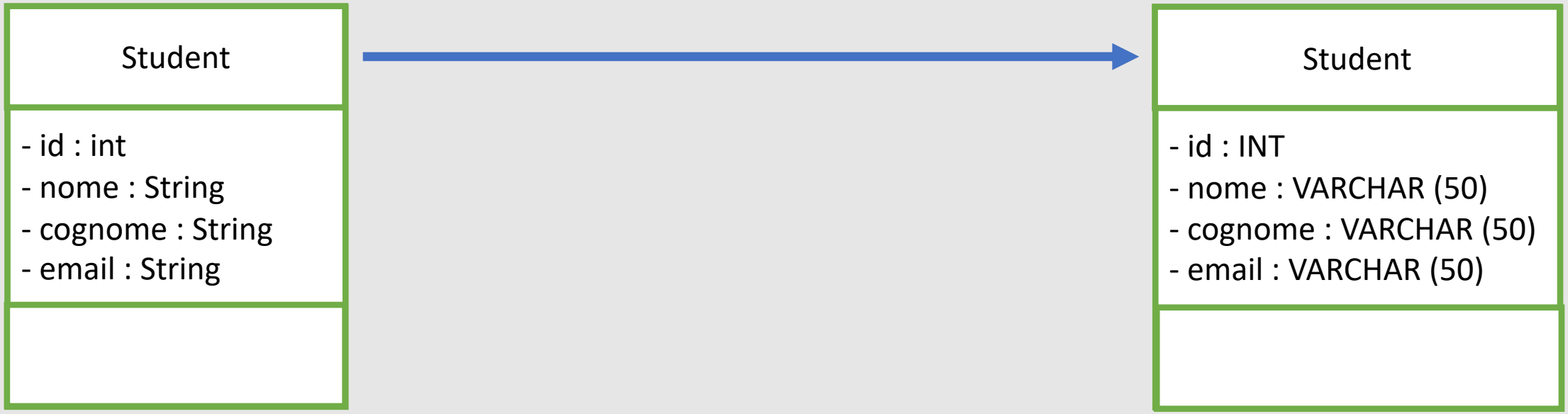
Annotare una classe Java

Una classe Entity è una classe Java che è mappata a una tabella del database



Annotare una classe Java

Una classe Entity non è altro che una classe con metodi get/set + annotazioni



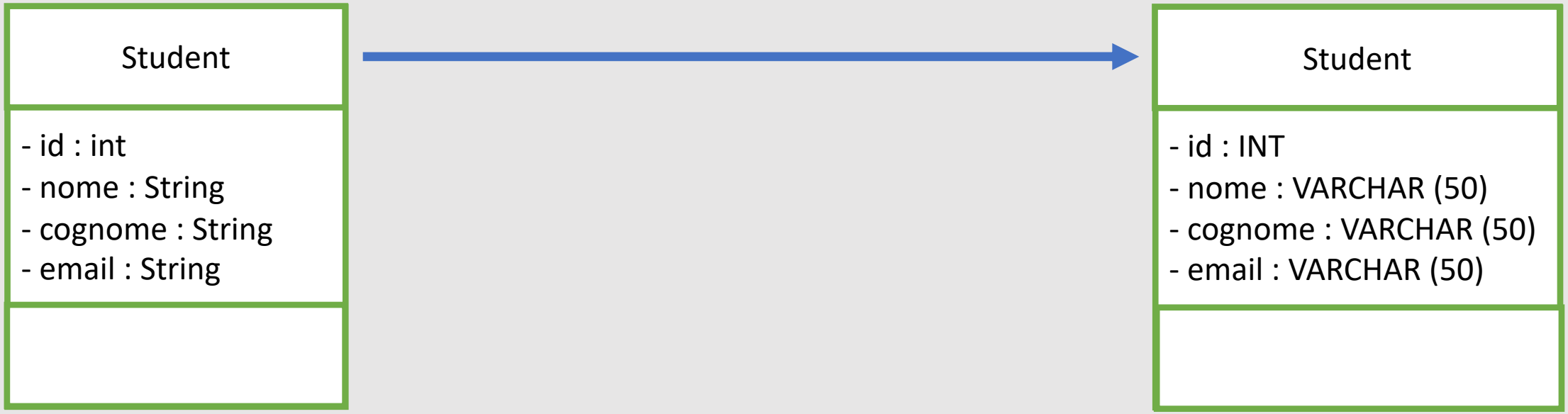
Annotare una classe Java

2 modi per mappare:

- File di configurazione xml (metodo vecchio)
- Annotazioni Java (metodo moderno)

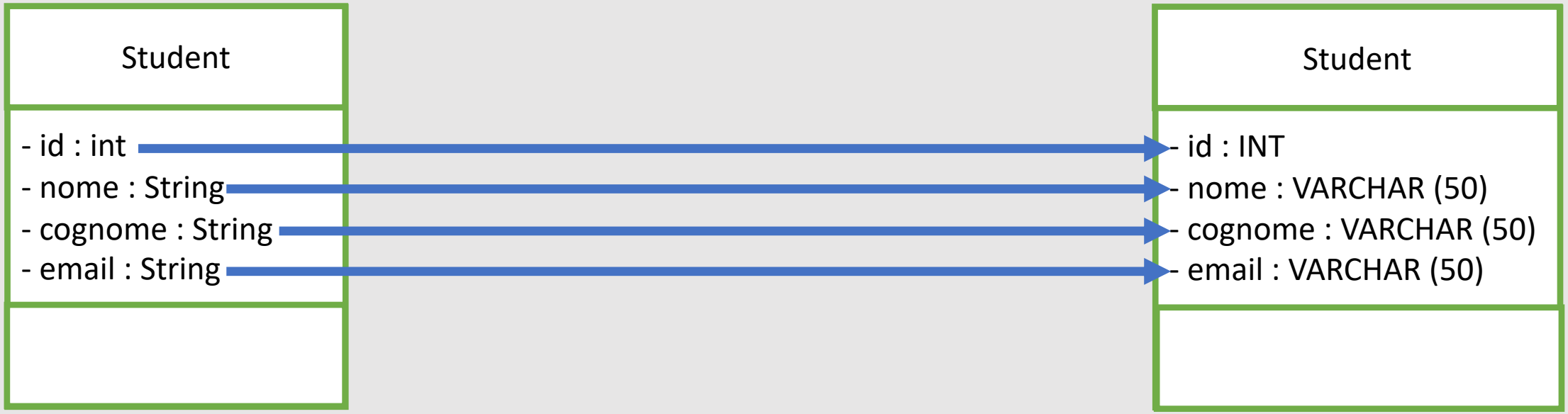
Annotare una classe Java

1. Mappare la classe alla tabella sul database



Annotare una classe Java

1. Mappare i campi alle colonne del database



Mappare la classe alla tabella

```
@Entity
@Table(name = "student")
public class Student {
  ...
}
```

- @Entity : dice che è un’entity
- name="student” : nome della tabella (facoltativo se il nome della classe è uguale a quello della tabella)

Mappare i campi alle colonne

```java
@Id
@Column(name = "id")
private int id;

@Column(name = "first_name")
private String firstName;
```

- @Id : chiave primaria
- name=”id” : nome della Colonna
- *name* e campo possono essere diversi, basta che il 1° mappi la colonna. Se uguali non è necessario specificare *name*

DEMO

SessionFactory

- Legge il file di configurazione di hibernate
- Ottiene una connessione e crea oggetti di sessione
- Oggetto pesante = crei una volta e lo riutilizzi

SessionFactory

```
SessionFactory factory = new Configuration()
                            .configure("hibernate.cfg.xml")
                            .addAnnotatedClass(Student.class)
                            .buildSessionFactory();
```

- .configure() : prende come default "hibernate.cfg.xml"
- .configure("hibernate.cfg.xml")

Session

- Instaura una connessione JDBC
- Serve per salvare/recuperate oggetti
- Oggetto di breve durata
- Si recupera dalla SessionFactory

SessionFactory

```
Session session = factory.getCurrentSession();
```

- Recupera la sessione dalla SessionFactory

Salvare oggetto Java

```
session.beginTransaction();
session.save(student);
session.getTransaction().commit();
```

- session.beginTransaction() : creo una transazione
- session.save(student) : salvo l'oggetto su db (INSERT)
- session.getTransaction().commit() : committo l'operazione

DEMO

Q&A