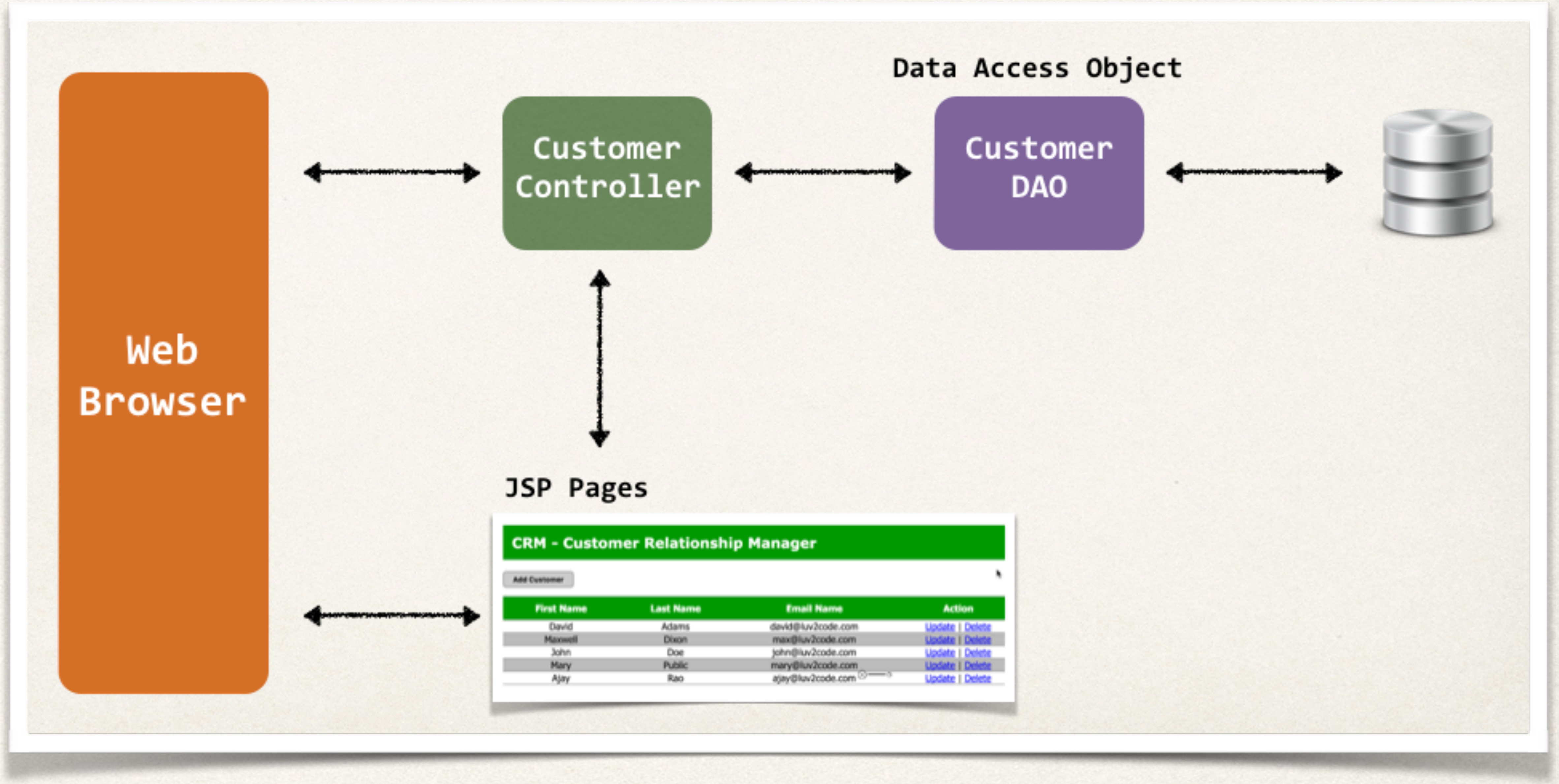
List Customers

Step-By-Step

1. Create **Customer.java**
2. Create **CustomerDAO.java**
    1. and **CustomerDAOImpl.java**
3. Create **CustomerController.java**
4. Create JSP page: **list-customers.jsp**



CRM - Customer Relationship Manager

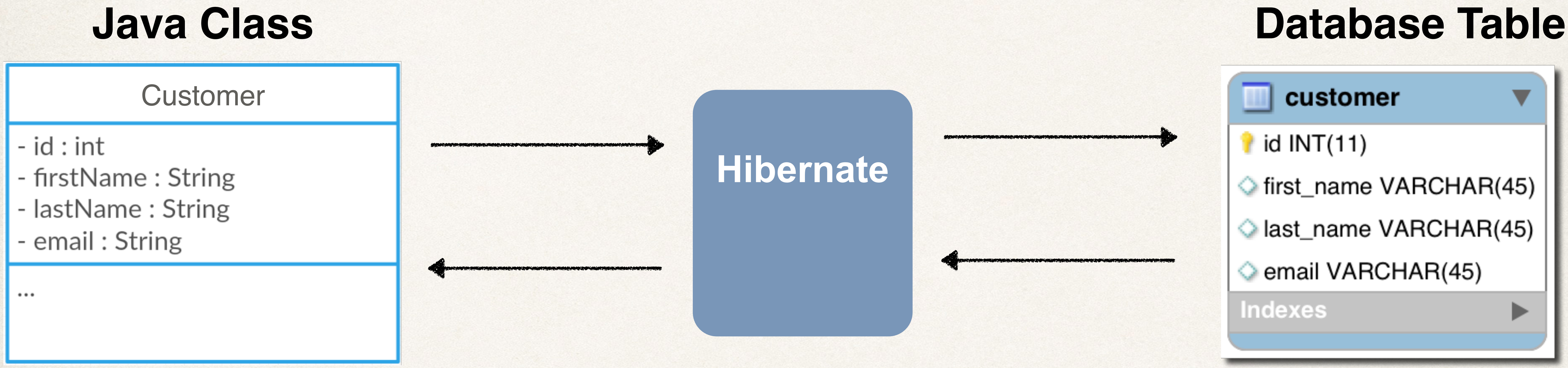
First Name	Last Name	Email Name	Action
David	Adams	david@luv2code.com	Update | Delete
Maxwell	Dixon	max@luv2code.com	Update | Delete
John	Doe	john@luv2code.com	Update | Delete
Mary	Public	mary@luv2code.com	Update | Delete
Ajay	Rao	ajay@luv2code.com	Update | Delete

Hibernate Terminology - Refresh

Entity Class

Java class that is mapped to a database table

Object-to-Relational Mapping (ORM)



Java Class

Customer

- id : int
- firstName : String
- lastName : String
- email : String

...

Database Table

customer

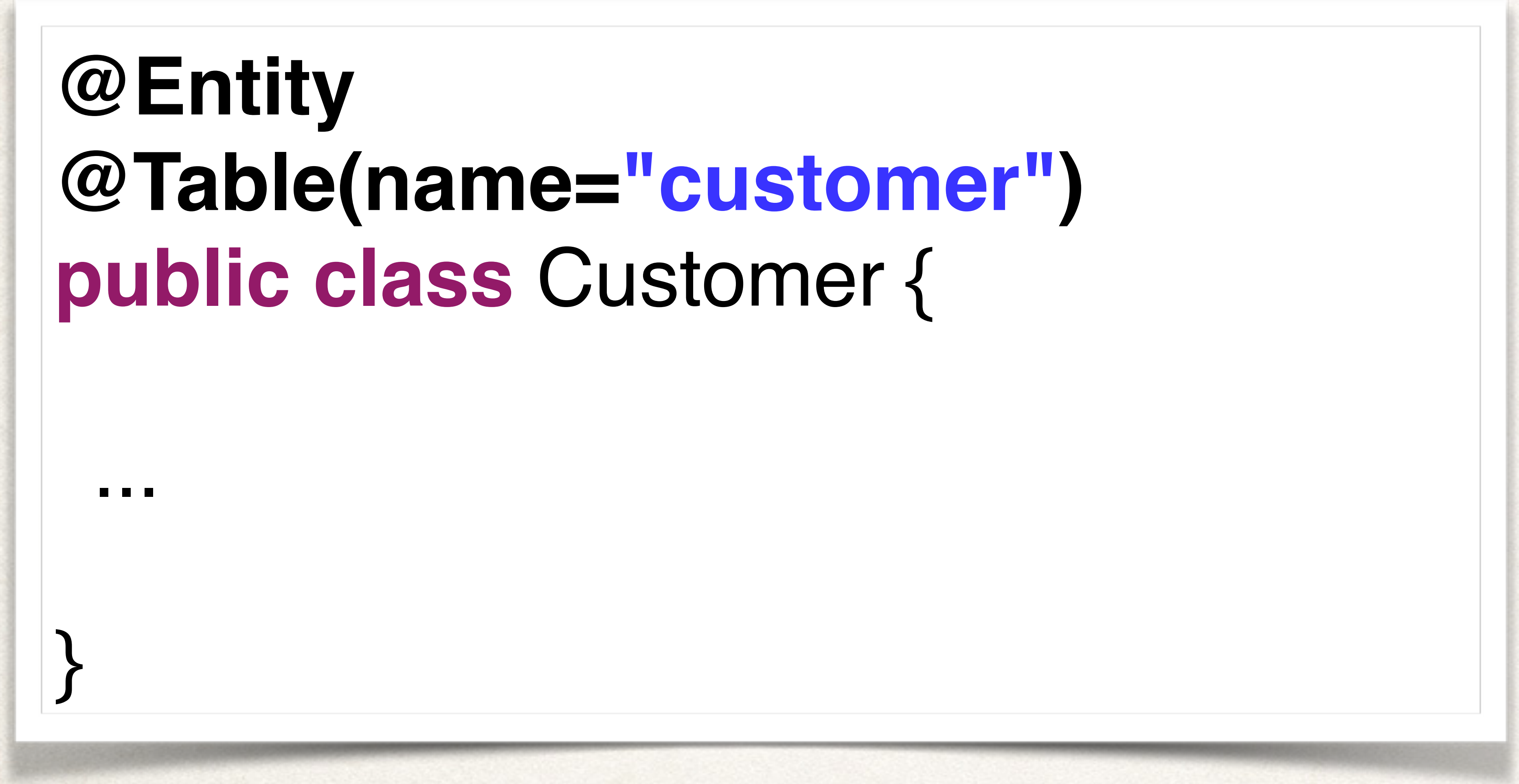
id INT(11)

first_name VARCHAR(45)

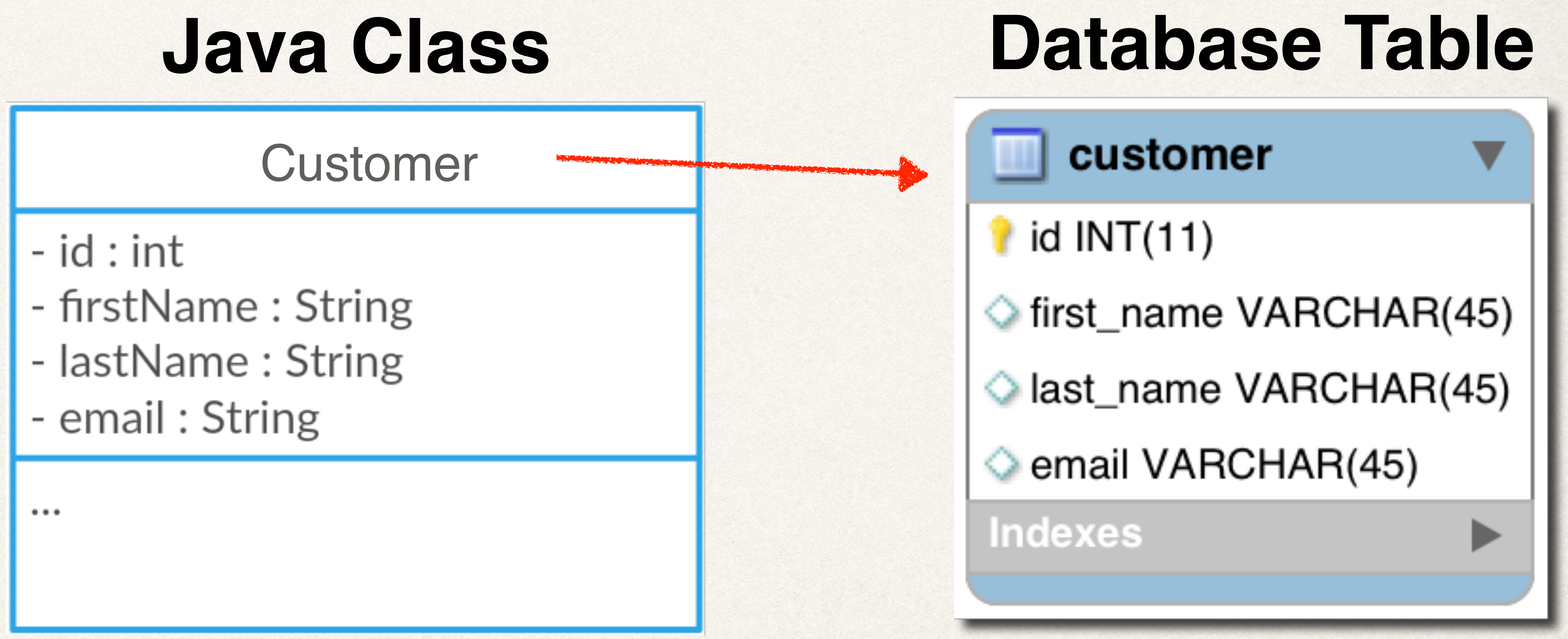
last_name VARCHAR(45)

email VARCHAR(45)

Step 1: Map class to database table



```
@Entity
@Table(name="customer")
public class Customer {

  ...

}
```



Java Class

Customer

- id : int
- firstName : String
- lastName : String
- email : String

...

Database Table

customer

id INT(11)

first_name VARCHAR(45)

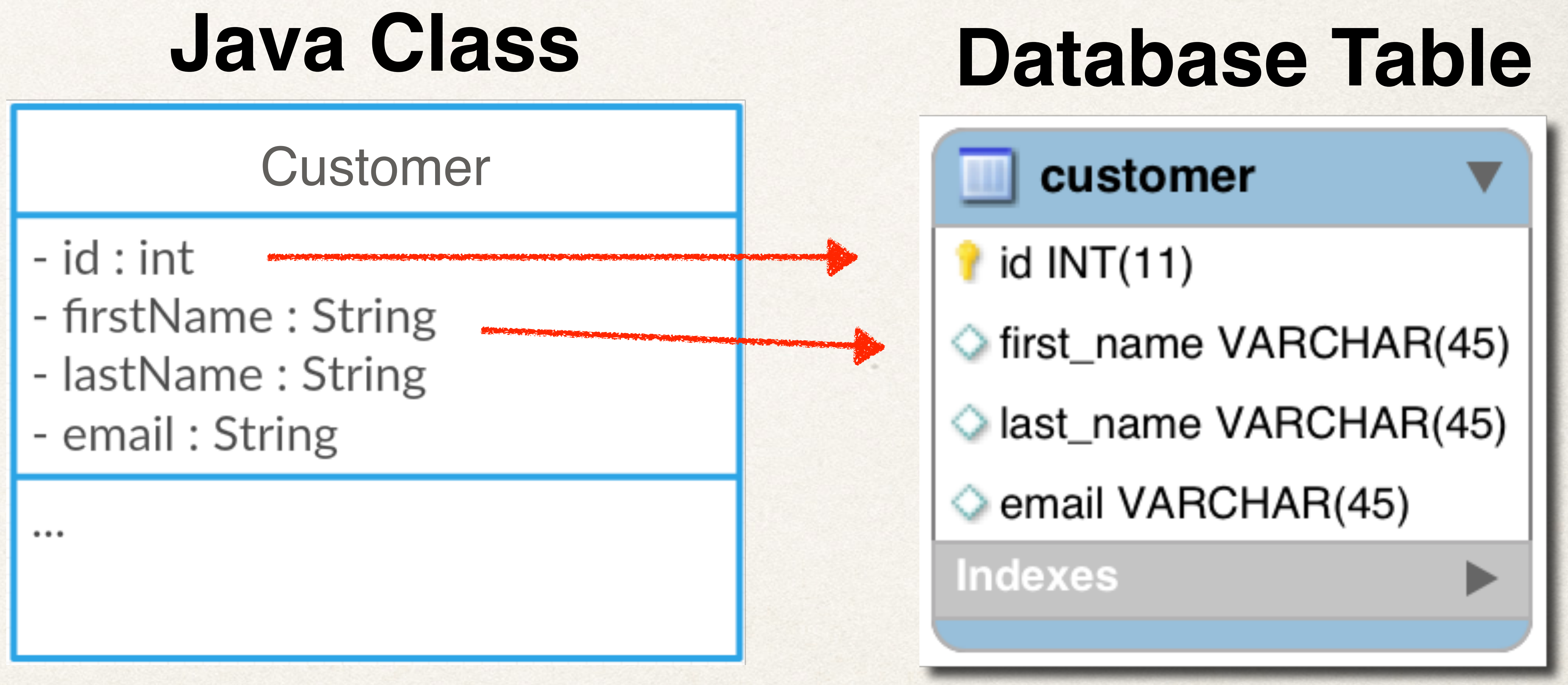
last_name VARCHAR(45)

email VARCHAR(45)

Indexes

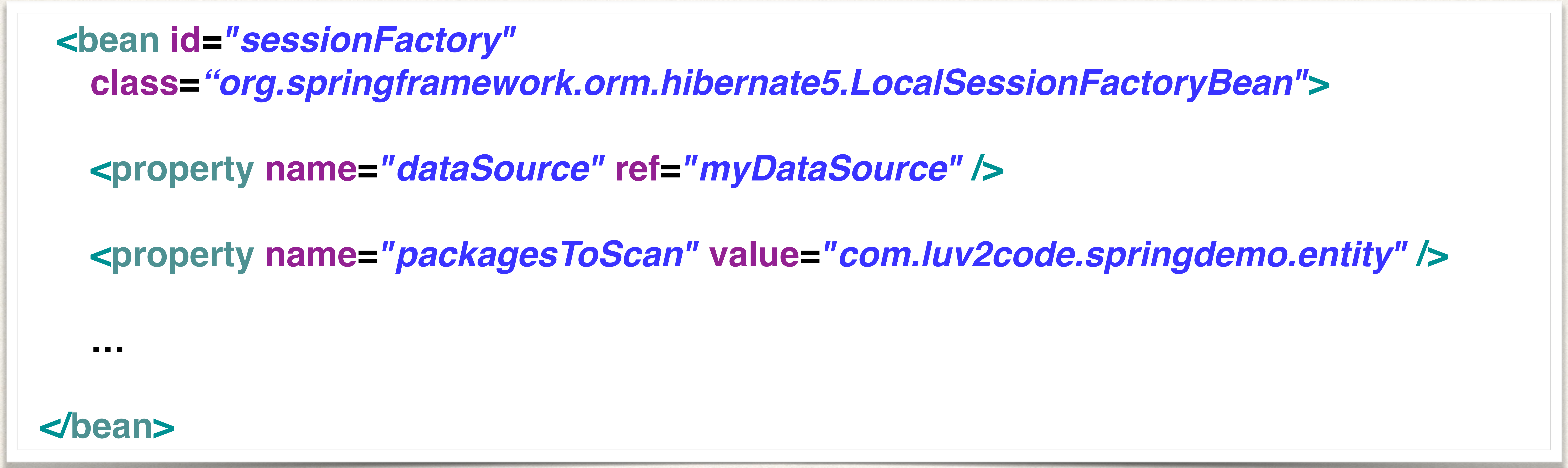
Step 2: Map fields to database columns

```
@Entity
@Table(name="customer")
public class Customer {

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

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



Entity Scanning

- Remember our Spring MVC config file?



```
<bean id="sessionFactory"
   class=“org.springframework.orm.hibernate5.LocalSessionFactoryBean">

   <property name="dataSource" ref="myDataSource" />

   <property name="packagesToScan" value="com.luv2code.springdemo.entity" />

   ...

</bean>
```