Cascade Delete

Foreign key column

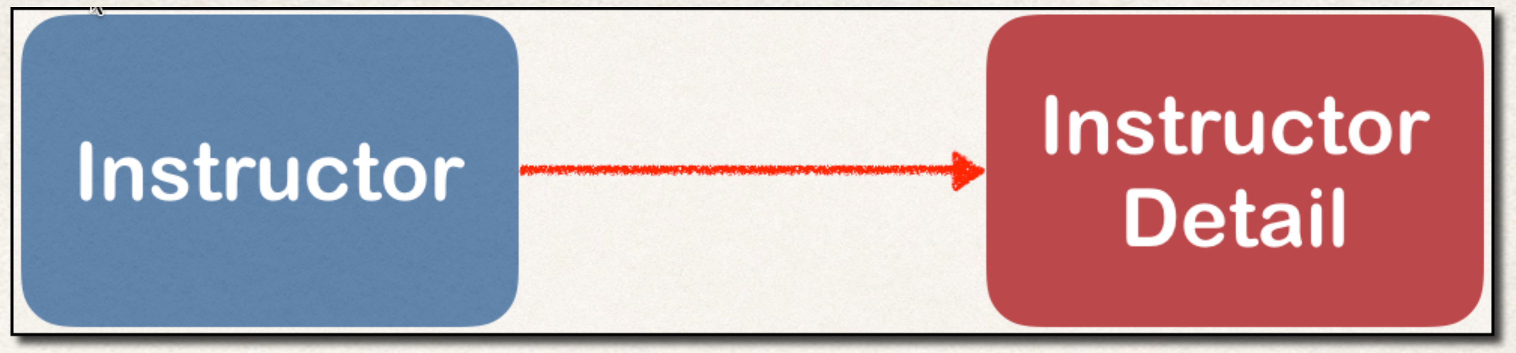
Table: instructor

id	first_name	last_name	instructor_detail_id
~~1~~	~~Chad~~	~~Darby~~	~~100~~
2	Madhu	Patel	200

Table: instructor_detail

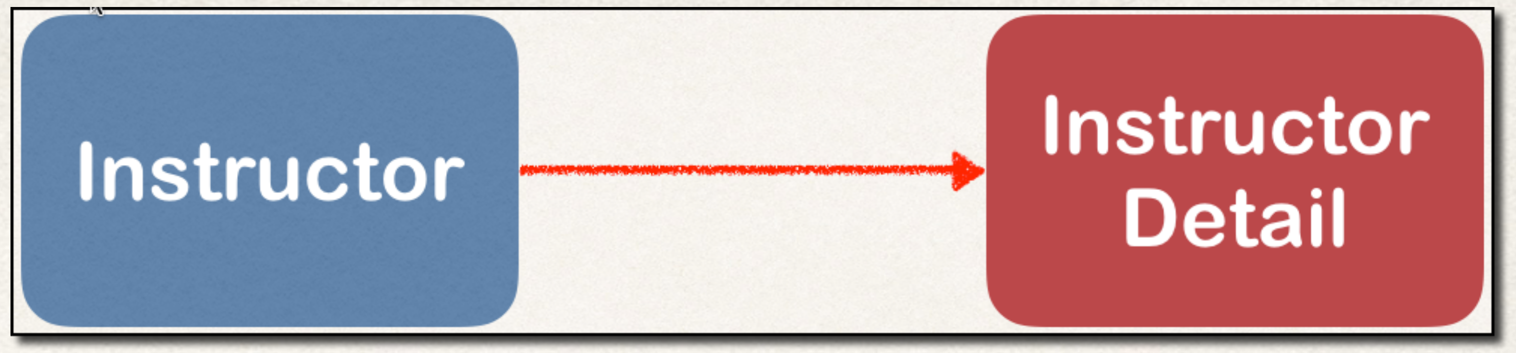
id	youtube_channel	hobby
~~100~~	~~www.luv2code.com/youtube~~	~~Luv 2 Code!!!~~
200	www.youtube.com	Guitar

@OneToOne - Cascade Types



Cascade Type	Description
PERSIST	If entity is persisted / saved, related entity will also be persisted
REMOVE	If entity is removed / deleted, related entity will also be deleted
REFRESH	If entity is refreshed, related entity will also be refreshed
DETACH	If entity is detached (not associated w/ session), then related entity will also be detached
MERGE	If entity is merged, then related entity will also be merged
ALL	All of above cascade types

Configure Cascade Type



```
@Entity
@Table(name="instructor")
public class Instructor {
   …

   @OneToOne(cascade=CascadeType.ALL)
   @JoinColumn(name="instructor_detail_id")
   private InstructorDetail instructorDetail;


   …
   // constructors, getters / setters
}
```

By default, no operations are cascaded.

Configure Multiple Cascade Types

```
@OneToOne(cascade={CascadeType.DETACH,
                   CascadeType.MERGE,
                   CascadeType.PERSIST,
                   CascadeType.REFRESH,
                   CascadeType.REMOVE})
```

Step 4: Create Main App

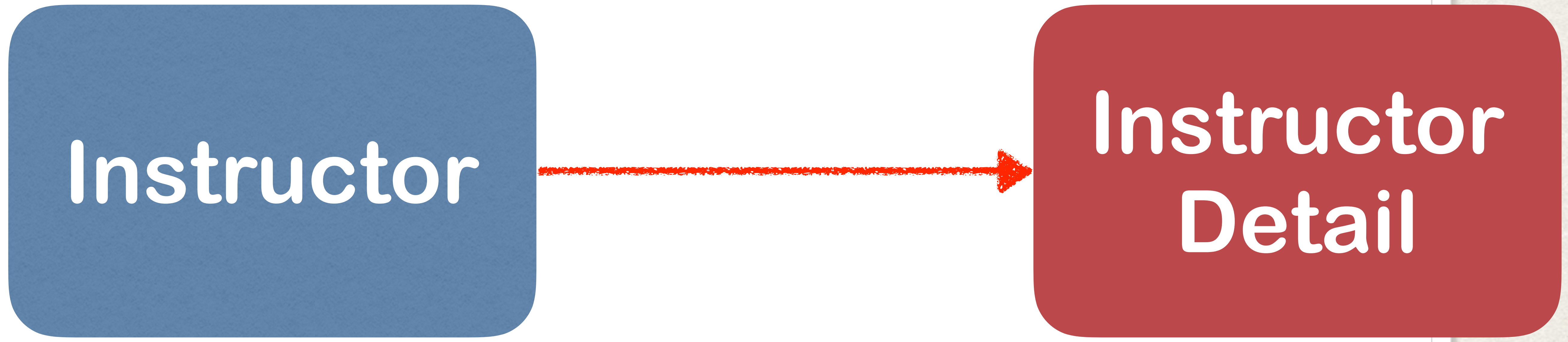
```
public static void main(String[] args) {
        ...










        ...
}
```

Step 4: Create Main App

```
public static void main(String[] args) {
        …
        // create the objects
        Instructor tempInstructor = new Instructor("Chad", "Darby", "darby@luv2code.com");

        InstructorDetail tempInstructorDetail =
                new InstructorDetail("http://www.luv2code.com/youtube", "Luv 2 code!!!");




        …
}
```

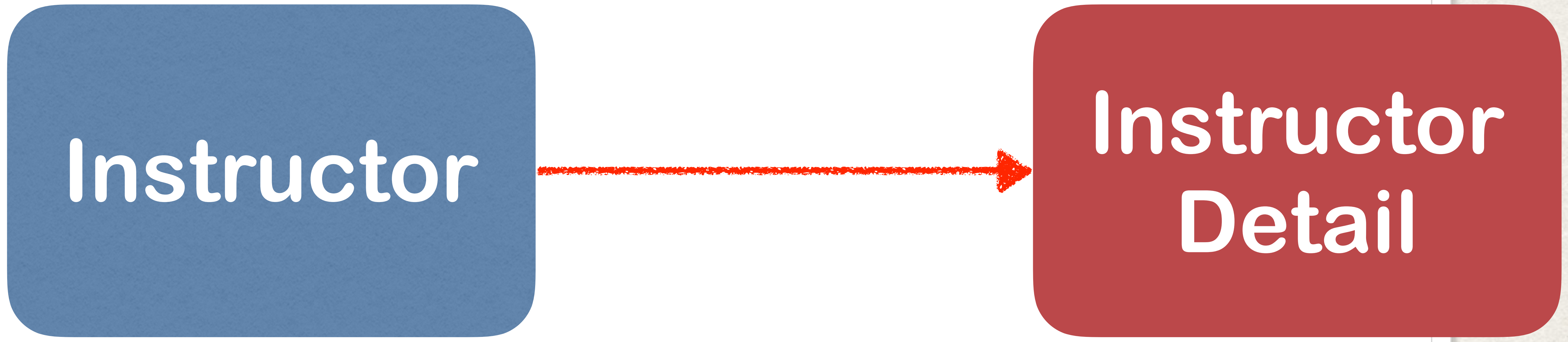
Step 4: Create Main App

```
public static void main(String[] args) {
        ...
        // create the objects
        Instructor tempInstructor = new Instructor("Chad", "Darby", "darby@luv2code.com");

        InstructorDetail tempInstructorDetail =
                new InstructorDetail(“http://www.luv2code.com/youtube", "Luv 2 code!!!");

        // associate the objects
        tempInstructor.setInstructorDetail(tempInstructorDetail);

        ...
}
```



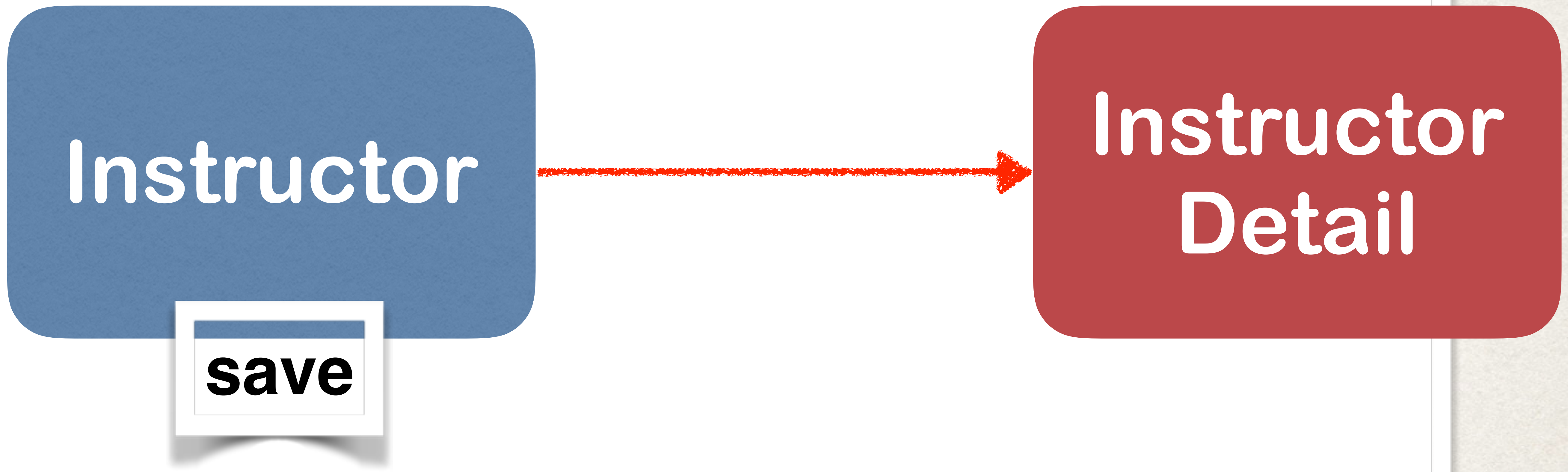
Step 4: Create Main App

```
public static void main(String[] args) {
        ...
        // create the objects
        Instructor tempInstructor = new Instructor("Chad", "Darby", "darby@luv2code.com");

        InstructorDetail tempInstructorDetail =
                new InstructorDetail("http://www.luv2code.com/youtube", "Luv 2 code!!!");

        // associate the objects
        tempInstructor.setInstructorDetail(tempInstructorDetail);

        // start a transaction
        session.beginTransaction();



        ...
}
```



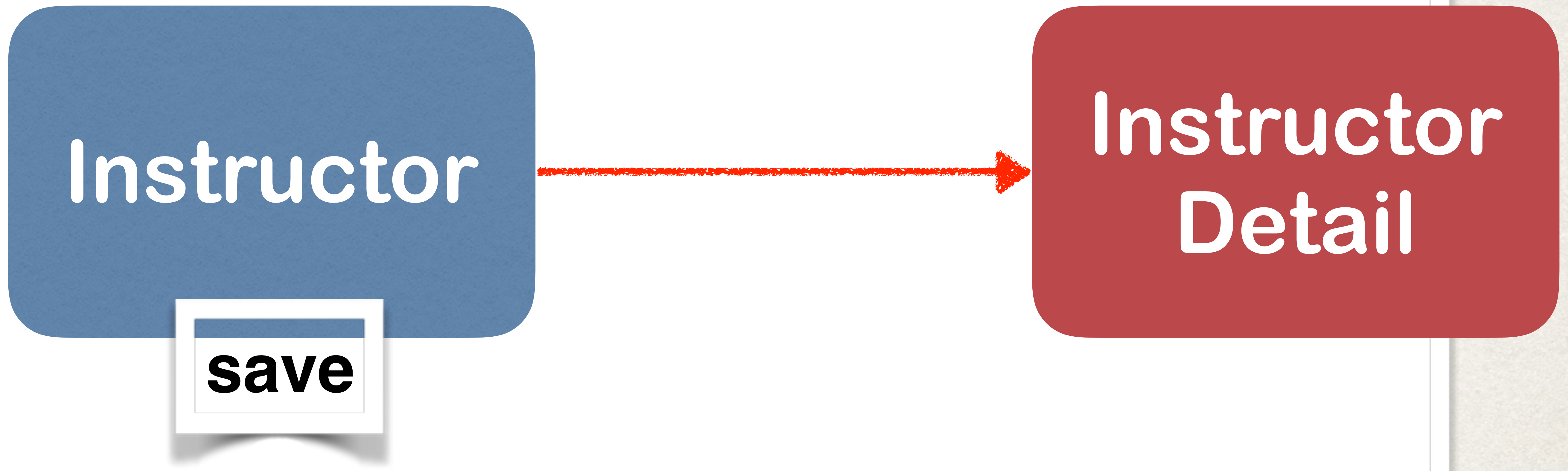
Step 4: Create Main App

```
public static void main(String[] args) {
        …
        // create the objects
        Instructor tempInstructor = new Instructor("Chad", "Darby", "darby@luv2code.com");

        InstructorDetail tempInstructorDetail =
                new InstructorDetail(“http://www.luv2code.com/youtube", "Luv 2 code!!!");

        // associate the objects
        tempInstructor.setInstructorDetail(tempInstructorDetail);

        // start a transaction
        session.beginTransaction();

        session.save(tempInstructor);



        …
}
```



Step 4: Create Main App

```
public static void main(String[] args) {
        …
        // create the objects
        Instructor tempInstructor = new Instructor("Chad", "Darby", "darby@luv2code.com");

        InstructorDetail tempInstructorDetail =
                new InstructorDetail("http://www.luv2code.com/youtube", "Luv 2 code!!!");

        // associate the objects
        tempInstructor.setInstructorDetail(tempInstructorDetail);

        // start a transaction
        session.beginTransaction();

        session.save(tempInstructor);

        // commit transaction
        session.getTransaction().commit();
        …
}
```


Instructor
save
Instructor Detail