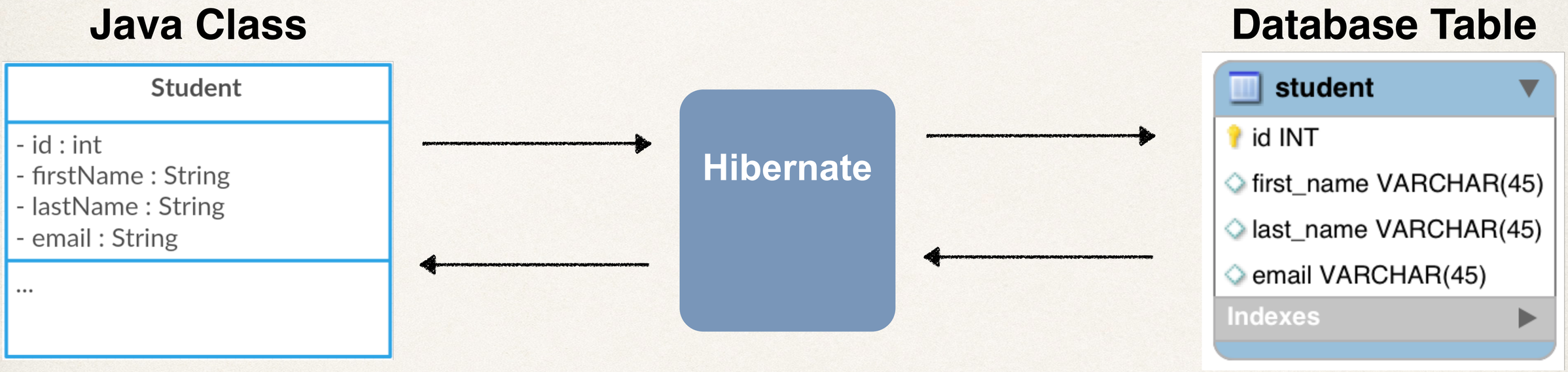
luv2code

Hibernate Advanced Mappings

Basic Mapping



Advanced Mappings

- In the database, you most likely will have
  - Multiple Tables
  - Relationships between Tables

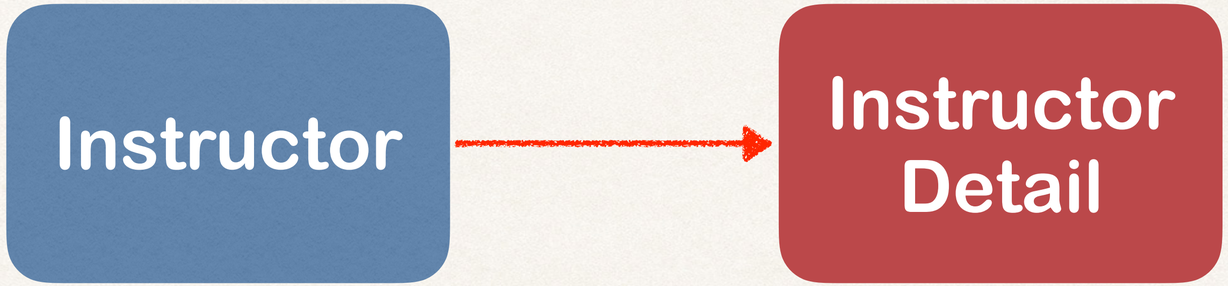
- Need to model this with Hibernate

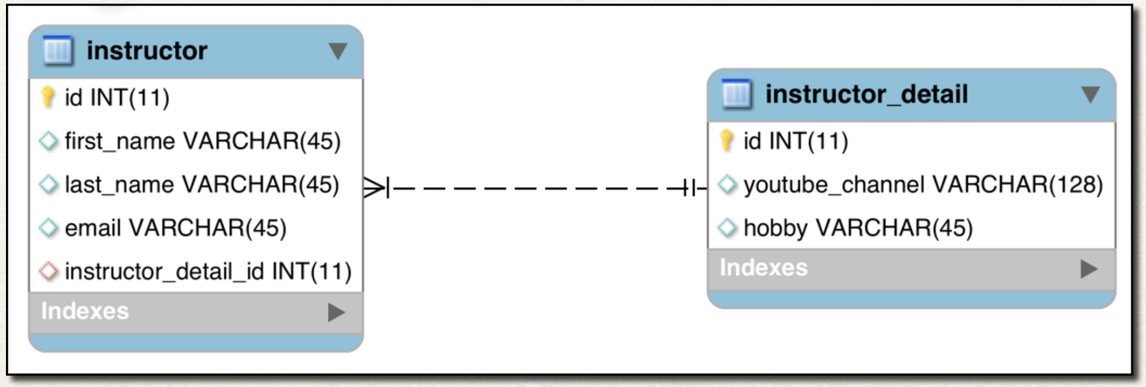
Advanced Mappings

- One-to-One
- One-to-Many, Many-to-One
- Many-to-Many

One-to-One Mapping

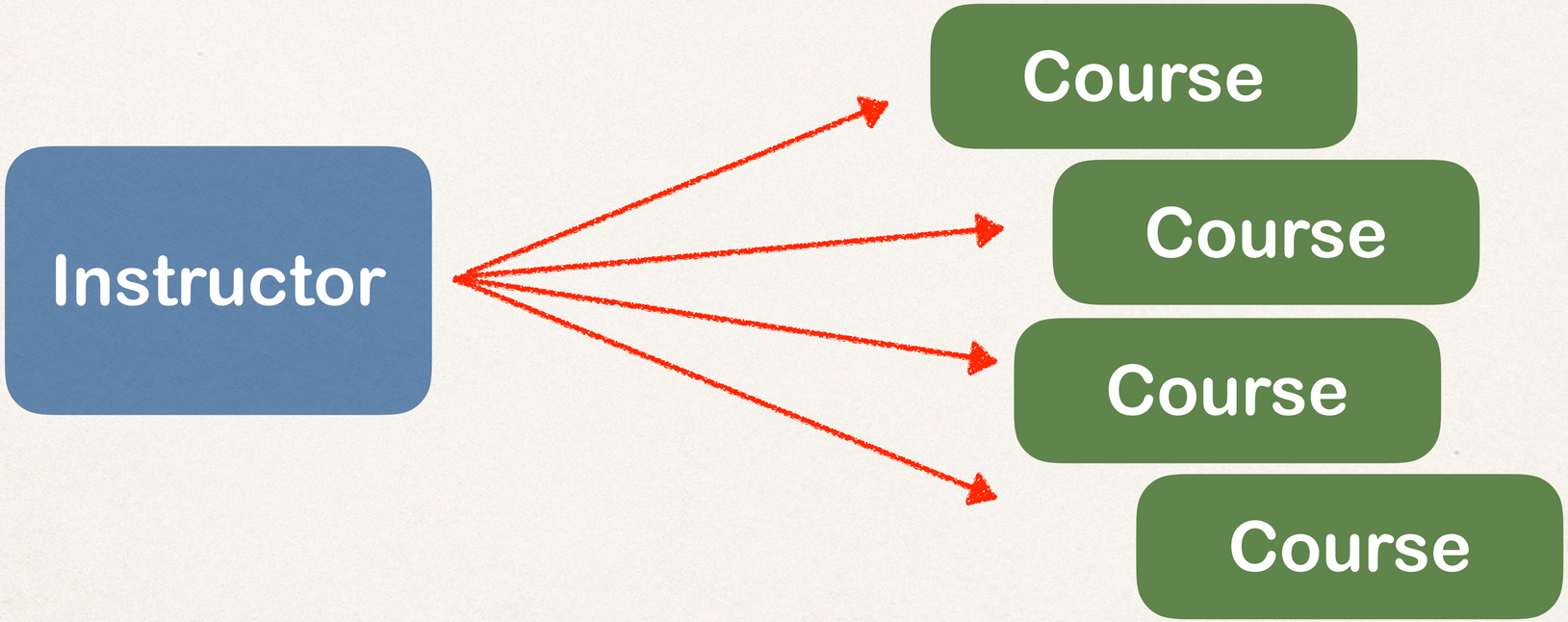
- An instructor can have an “instructor detail” entity
  - Similar to an “instructor profile”





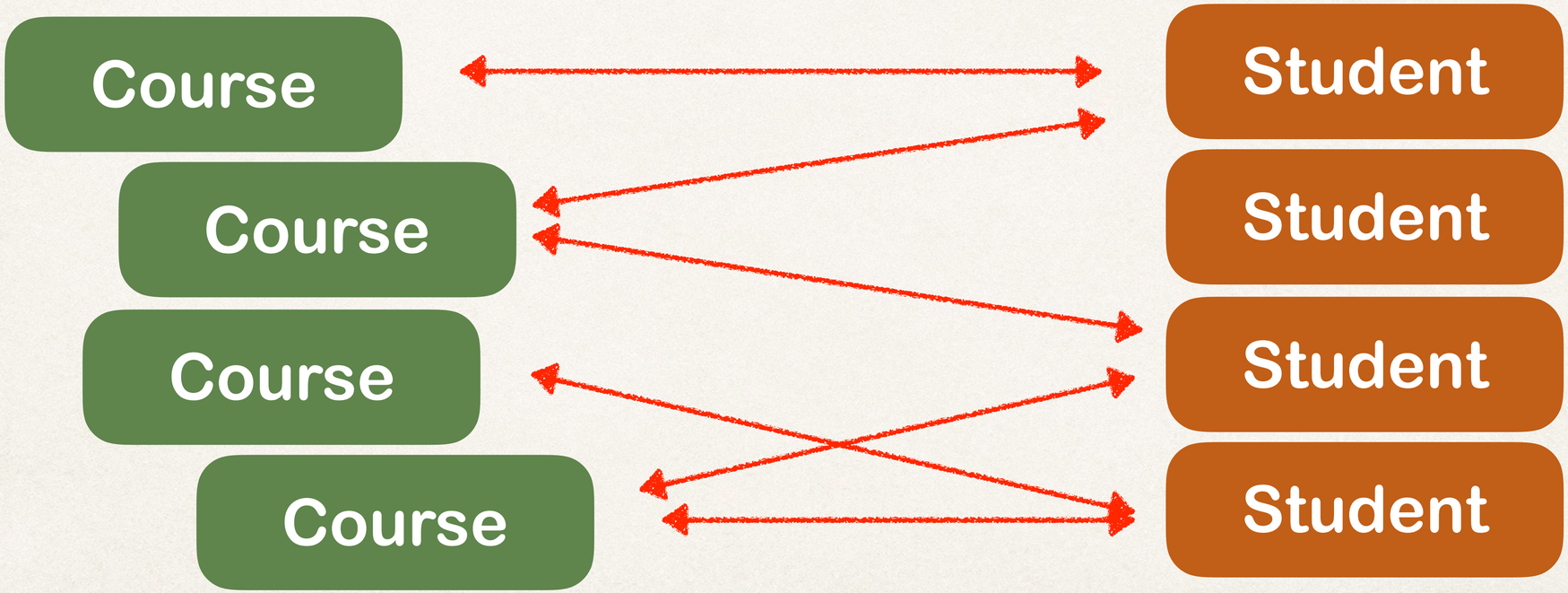
One-to-Many Mapping

- An instructor can have many courses



Many-to-Many Mapping

- A course can have many students
- A student can have many courses



Important Database Concepts

- Primary key and foreign key

- Cascade

Primary Key and Foreign Key

- Primary key: identify a unique row in a table

- Foreign key:
  - Link tables together
  - a field in one table that refers to primary key in another table

Foreign Key Example

Table: instructor

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

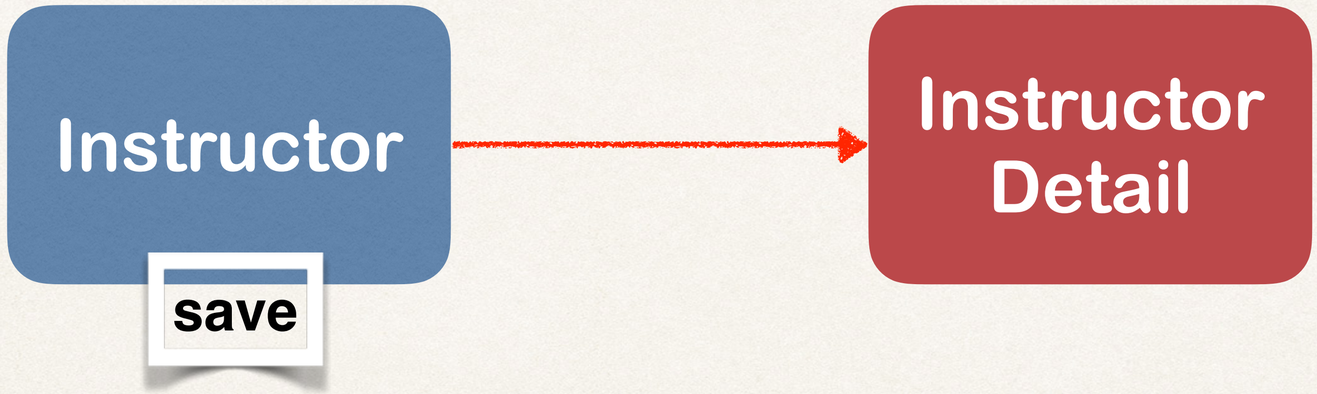
Table: instructor_detail

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

Cascade

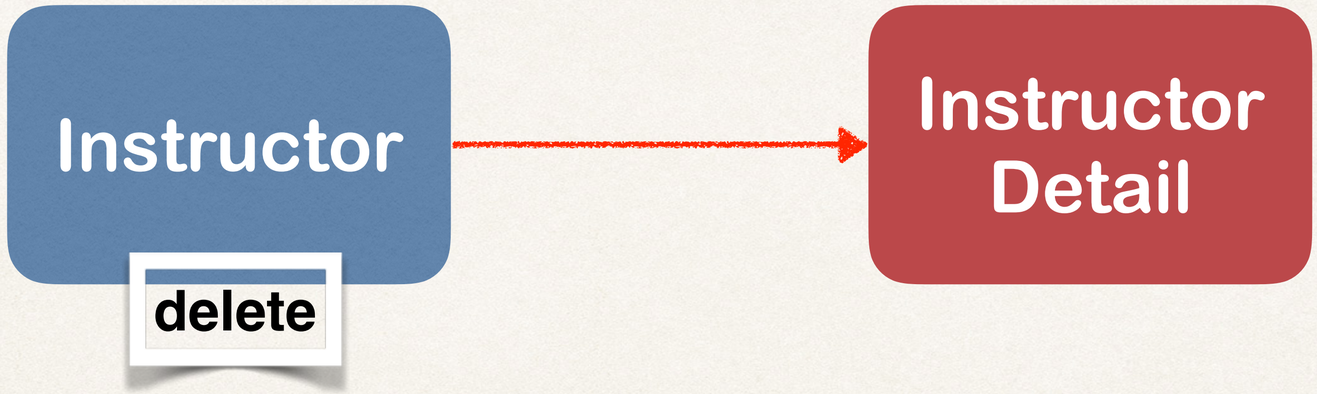
- You can **cascade** operations

- Apply the same operation to related entities



Cascade

- If we delete an **instructor**, we should also delete their **instructor_detail**
  - This is known as “CASCADE DELETE”



Cascade Delete

Foreign key column

Table: instructor

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

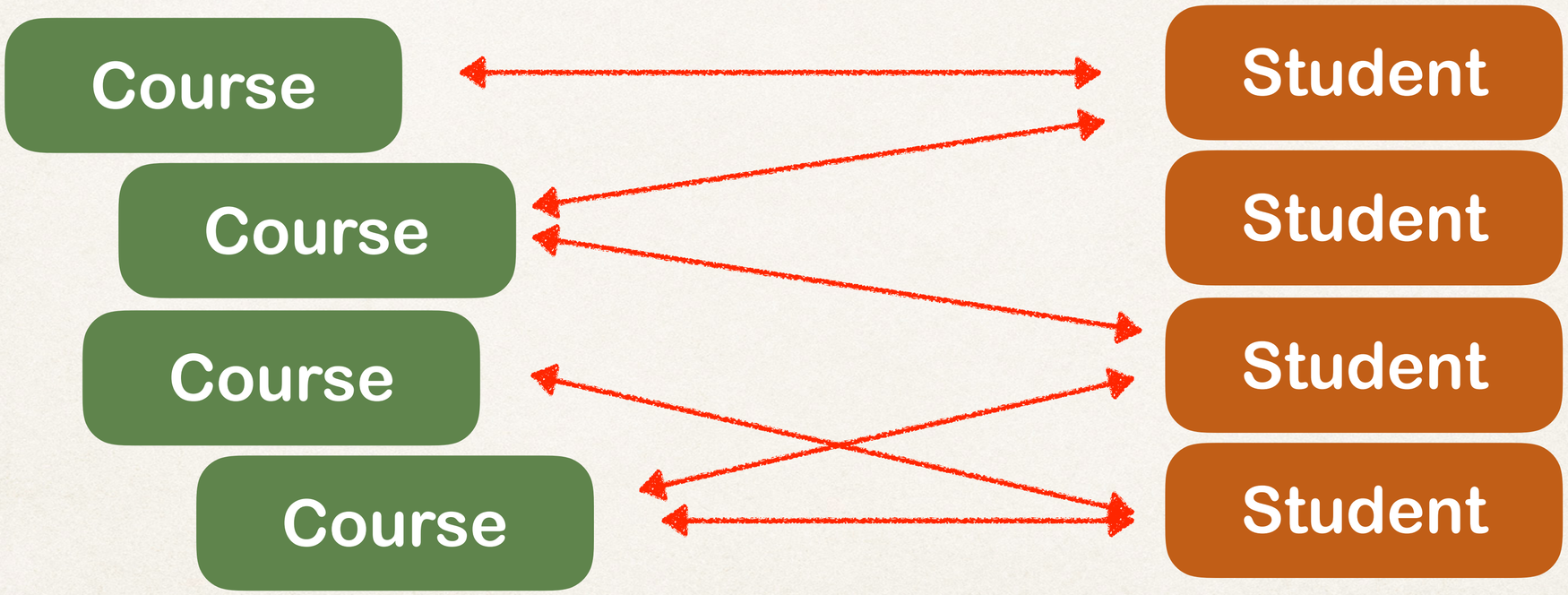
Table: instructor_detail

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

Cascade Delete

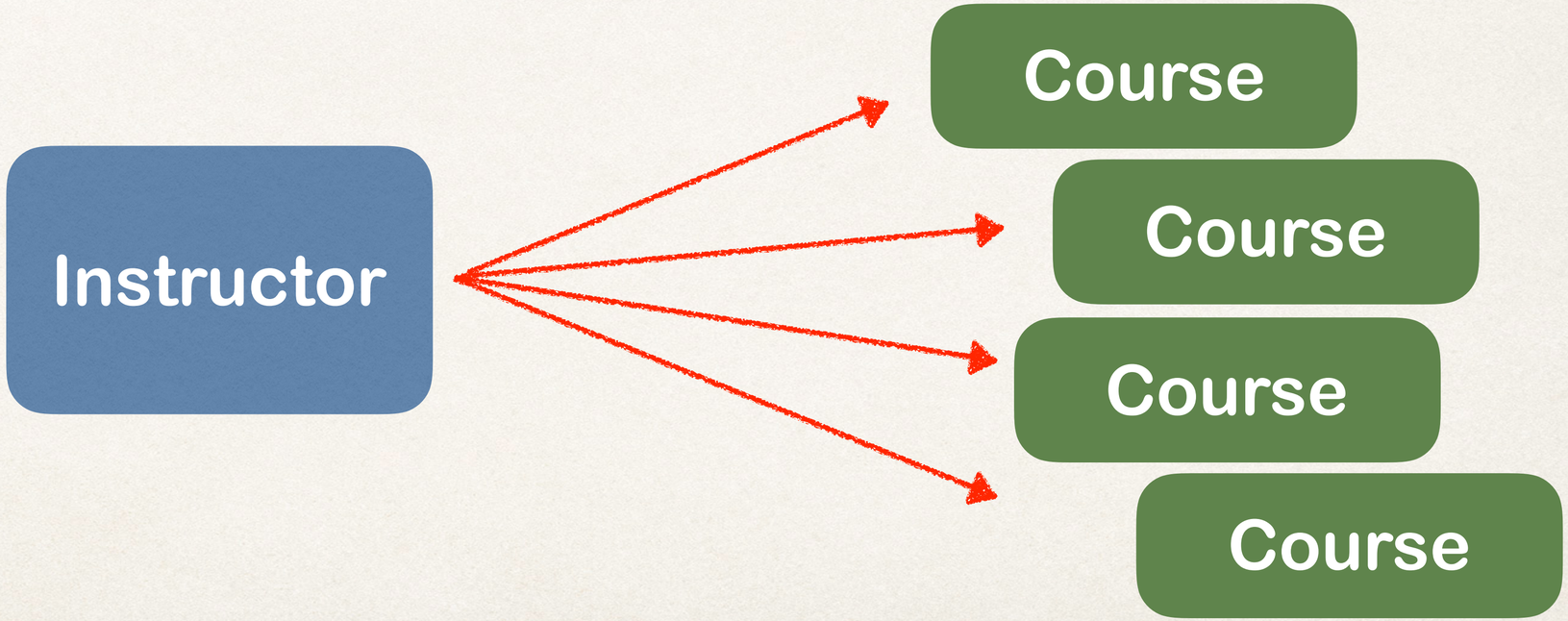
- Cascade delete depends on the use case
- Should we do cascade delete here???

Developer can configure cascading



Fetch Types: Eager vs Lazy Loading

- When we fetch / retrieve data, should we retrieve EVERYTHING?
  - **Eager** will retrieve everything
  - **Lazy** will retrieve on request



Uni-Directional

Bi-Directional