



# Active Record Associations

Cool Shit

Author: Torey Hickman & Casey Cumbow

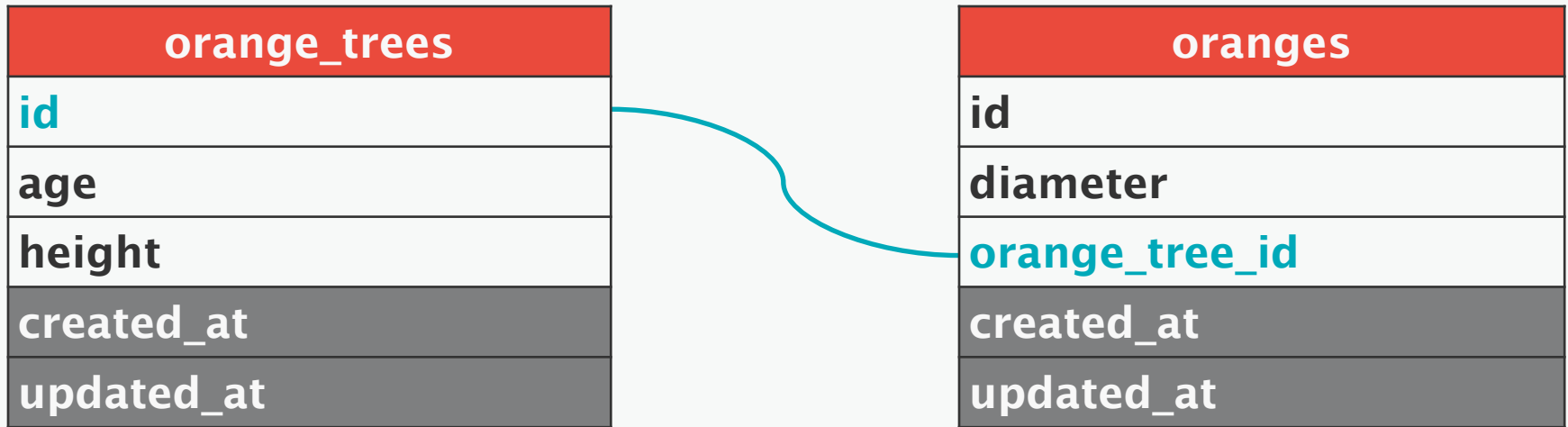
# Active Record Associations

- Recap on associations
- Spot the Errors drills

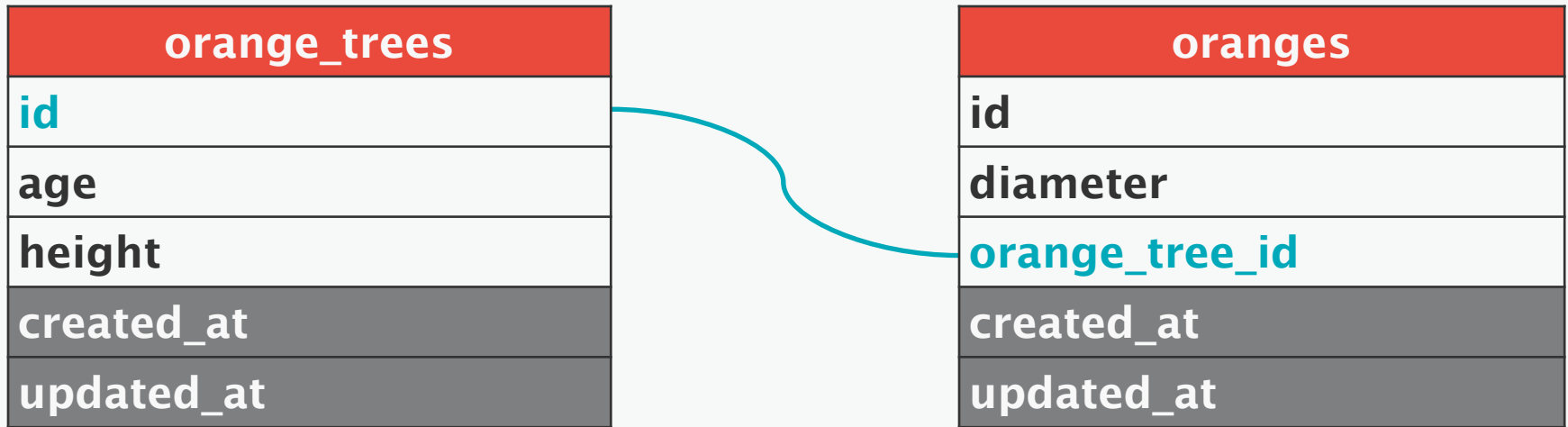
# Primary & Foreign Keys

- Primary keys and foreign keys provide data for joining tables

# Primary & Foreign Keys



# Primary & Foreign Keys



- The foreign key belongs on the table of the objects that are owned

# One To Many

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
end
```

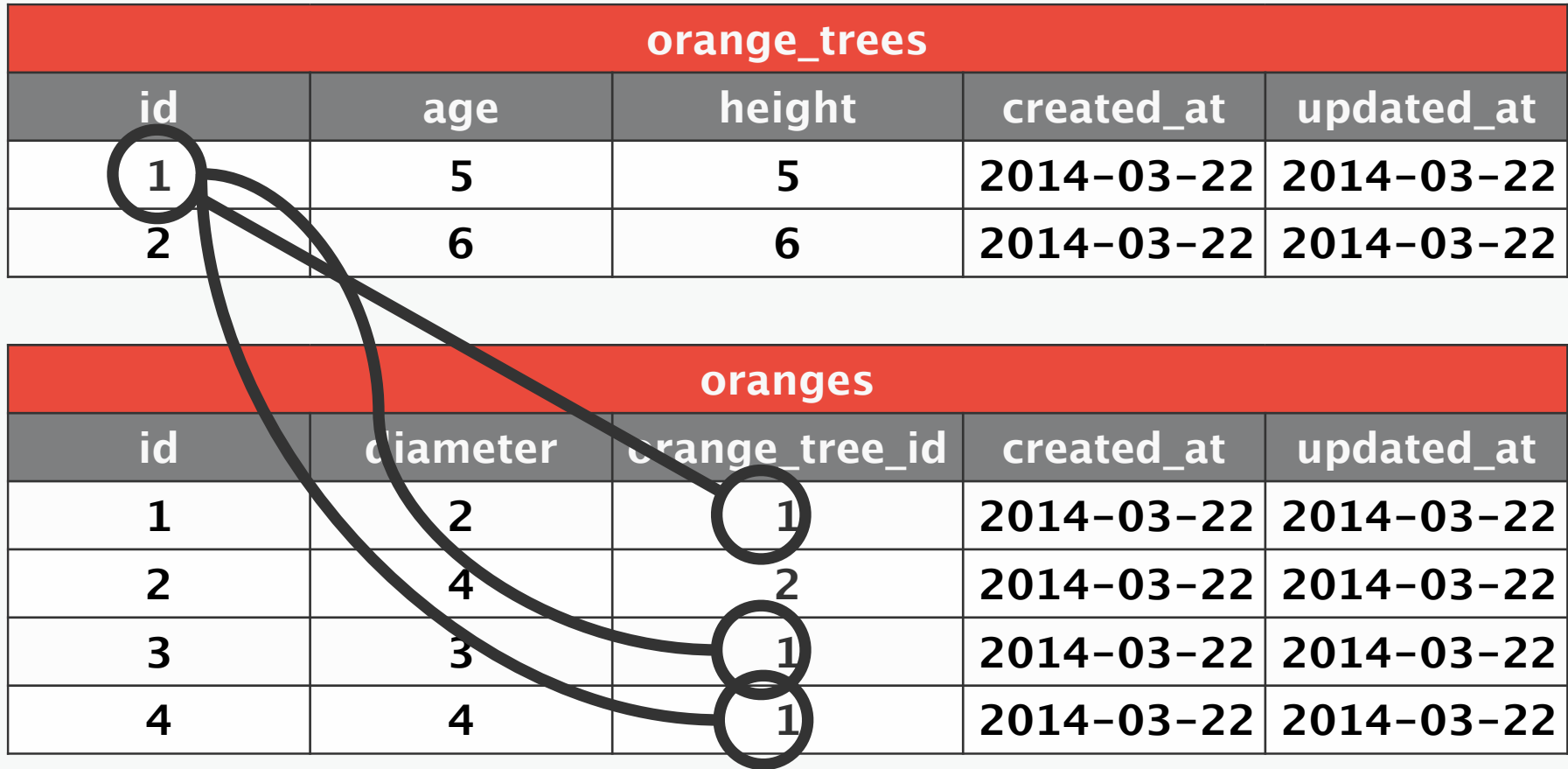
```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
end
```

# One To Many

orange_trees				
id	age	height	created_at	updated_at
1	5	5	2014-03-22	2014-03-22
2	6	6	2014-03-22	2014-03-22

oranges				
id	diameter	orange_tree_id	created_at	updated_at
1	2	1	2014-03-22	2014-03-22
2	4	2	2014-03-22	2014-03-22
3	3	1	2014-03-22	2014-03-22
4	4	1	2014-03-22	2014-03-22

# One To Many





# One To Many

orange_trees				
id	age	height	created_at	updated_at
1	5	5	2014-03-22	2014-03-22
2	6	6	2014-03-22	2014-03-22

oranges				
id	diameter	orange tree_id	created_at	updated_at
1	2	1	2014-03-22	2014-03-22
2	4	2	2014-03-22	2014-03-22
3	3	1	2014-03-22	2014-03-22
4	4	1	2014-03-22	2014-03-22

# One To Many

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
end
```

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
end
```

```
orange_tree_one = OrangeTree.first
```

```
orange_tree_one.oranges
# returns a collection of Orange objects
```

# One To Many

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
end
```

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
end
```

```
orange_one = Orange.first
```

```
orange_one.orange_tree
# returns OrangeTree object
```

# One To Many

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges

  def oranges
    Orange.where(orange_tree_id: self.id)
  end
end

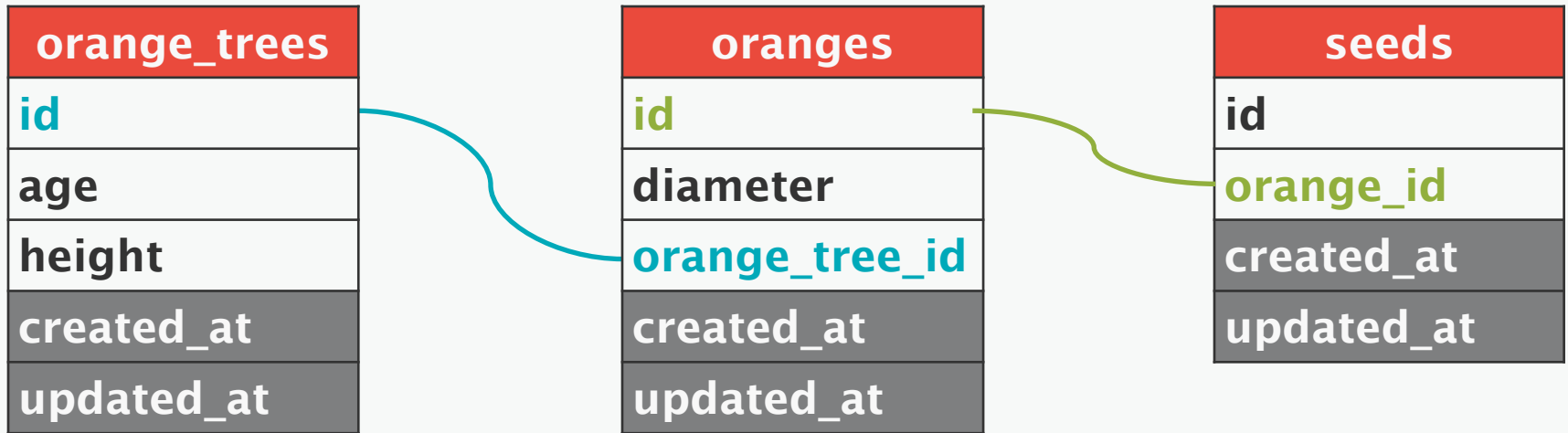
class Orange < ActiveRecord::Base
  belongs_to :orange_tree

  def orange_tree
    OrangeTree.find(self.orange_tree_id)
  end
end
```

# Has Many Through

- What if oranges have seeds?
- How would a tree have seeds?

# Has Many Through



# Has Many Through

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

```
class Seed < ActiveRecord::Base
  belongs_to :orange
end
```

# Has Many Through

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

```
class Seed < ActiveRecord::Base
  belongs_to :orange
end
```



# Has Many Through


```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

```
class Seed < ActiveRecord::Base
  belongs_to :orange
end
```

# Has Many Through

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```




```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

```
class Seed < ActiveRecord::Base
  belongs_to :orange
end
```

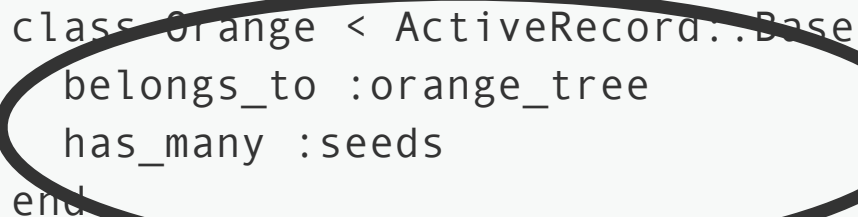
# Has Many Through

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```



The diagram illustrates the 'has\_many through' relationship. A line connects the `:seeds` association in the `OrangeTree` class to the `:oranges` association in the `Orange` class, indicating that `OrangeTree` has many seeds through its association with `Orange`.

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

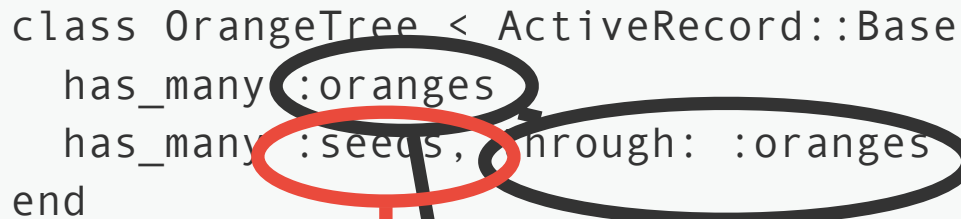


The diagram illustrates the 'has\_many' relationship. A line connects the `:seeds` association in the `Orange` class to the `:seeds` association in the `Seed` class, indicating that `Orange` has many seeds.


```
class Seed < ActiveRecord::Base
  belongs_to :orange
end
```

# Has Many Through

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```



```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```



```
class Seed < ActiveRecord::Base
  belongs_to :orange
end
```

# Has Many Through

orange_trees				
id	age	height	created_at	updated_at
1	5	5	2014-03-22	2014-03-22
2	6	6	2014-03-22	2014-03-22

oranges	
id	orange_tree_id
1	1
2	2
3	2
4	2

seeds	
id	orange_id
1	1
2	3
3	2
4	1

# Has Many Through

orange_trees				
id	age	height	created_at	updated_at
1	5	5	2014-03-22	2014-03-22
2	6	6	2014-03-22	2014-03-22

oranges	
id	orange_tree_id
1	1
2	2
3	2
4	2

seeds	
id	orange_id
1	1
2	3
3	2
4	1

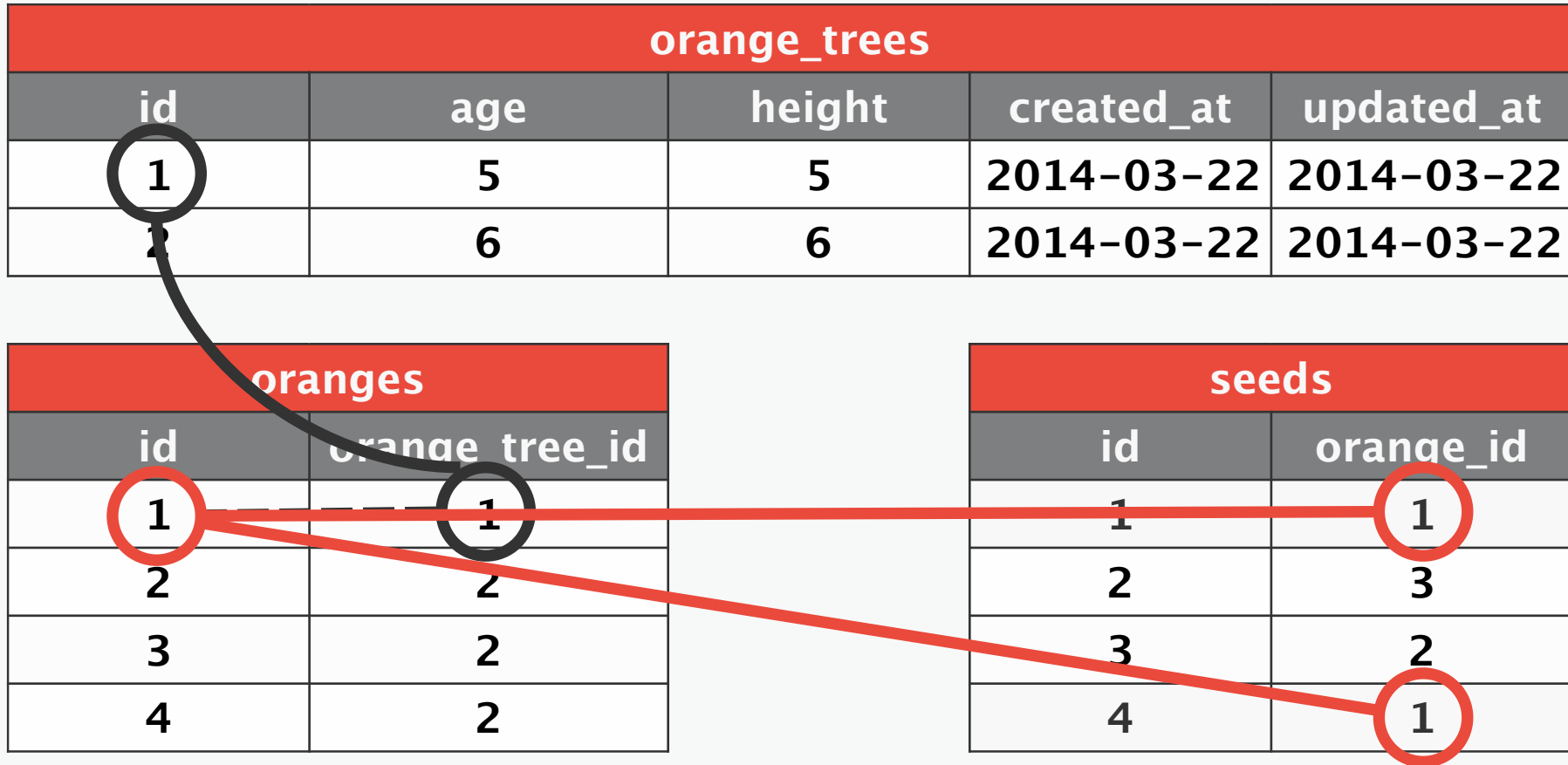
# Has Many Through

orange_trees				
id	age	height	created_at	updated_at
1	5	5	2014-03-22	2014-03-22
2	6	6	2014-03-22	2014-03-22

oranges	
id	orange_tree_id
1	1
2	2
3	2
4	2

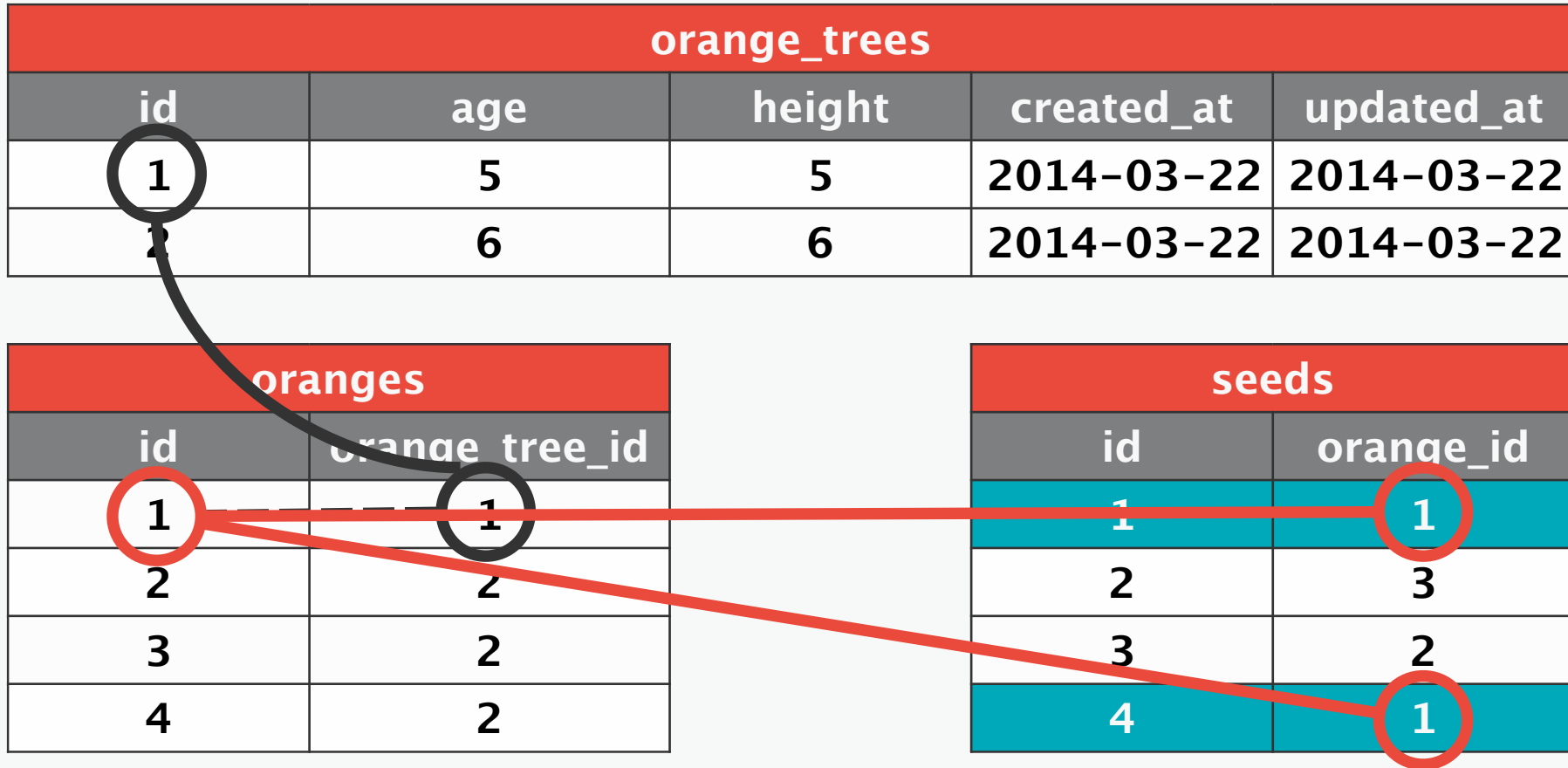
seeds	
id	orange_id
1	1
2	3
3	2
4	1

# Has Many Through





# Has Many Through



# Active Record Conventions

- ActiveRecord makes **ASSUMPTIONS** (like we saw earlier)
- **Tip:** write out with your words what AR assumes when you write an association

# Active Record Conventions

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

# Active Record Conventions

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

# Active Record Conventions

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

- There is an Orange class
- Table oranges has foreign key orange\_tree\_id

# Active Record Conventions

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

- There is an **Orange** class
- Table oranges has foreign key orange\_tree\_id

# Active Record Conventions

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

- There is an **Orange** class
- Table oranges has foreign key **orange\_tree\_id**

# Active Record Conventions

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```



# Active Record Conventions

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

- The oranges association works
- Orange class has an association with seed(s)

# Active Record Conventions

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

- The **oranges association** works
- Orange class has an association with seed(s)

# Active Record Conventions

```
class OrangeTree < ActiveRecord::Base
  has_many :oranges
  has_many :seeds, through: :oranges
end
```

- The **oranges** association works
- Orange class has an association with **seed(s)**

# Active Record Conventions

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

# Active Record Conventions

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

# Active Record Conventions

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

- There is an OrangeTree class
- Table oranges has foreign key orange\_tree\_id

# Active Record Conventions

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

- There is an **OrangeTree** class
- Table oranges has foreign key **orange\_tree\_id**

# Active Record Conventions

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

- There is an **OrangeTree** class
- Table oranges has foreign key **orange\_tree\_id**



# Active Record Conventions

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

# Active Record Conventions

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

- There is a Seed class
- Table seeds has foreign\_key orange\_id

# Active Record Conventions

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

- There is a **Seed** class
- Table seeds has foreign\_key orange\_id

# Active Record Conventions

```
class Orange < ActiveRecord::Base
  belongs_to :orange_tree
  has_many :seeds
end
```

- There is a **Seed** class
- Table seeds has foreign\_key **orange\_id**

# Spot the Errors

- What's wrong with these association?

# Spot the Errors



# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree
end
```



# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree
end
```





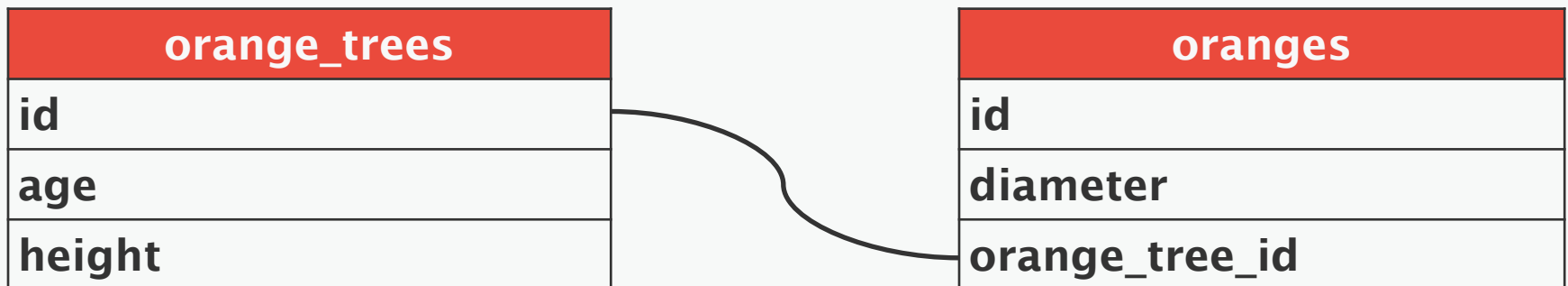
# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges,
    foreign_key: :tree_id
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree,
    foreign_key: :tree_id
end
```



# Spot the Errors



# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :fruits
end
```

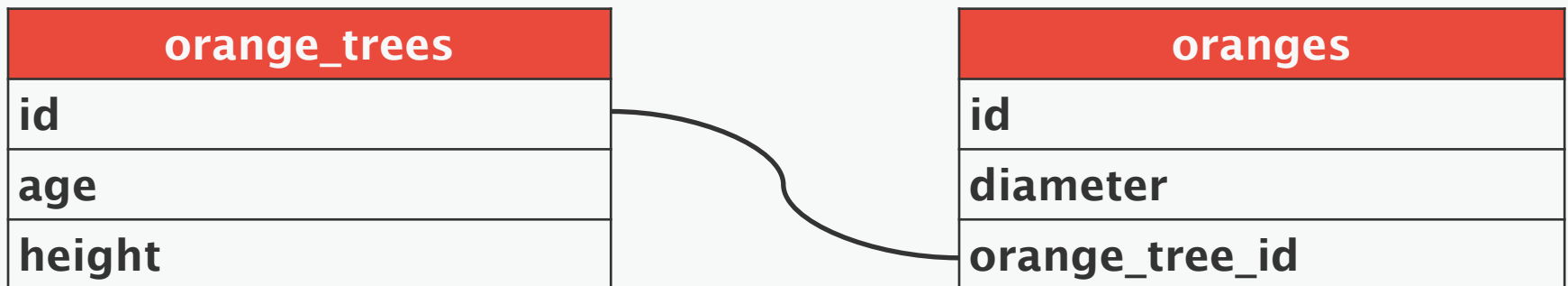
```
class Orange < AR::Base
  belongs_to :orange_tree
end
```



# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :fruits
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree
end
```



# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :fruits,
    class_name: "Orange"
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree
end
```



# Spot the Errors



# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
end
```

```
class Orange < AR::Base
  belongs_to :tree
end
```



# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
end
```

```
class Orange < AR::Base
  belongs_to :tree
end
```





# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
end
```

```
class Orange < AR::Base
  belongs_to :tree,
    class_name: "OrangeTree"
end
```



# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
end
```

```
class Orange < AR::Base
  belongs_to :tree,
    class_name: "OrangeTree"
end
```



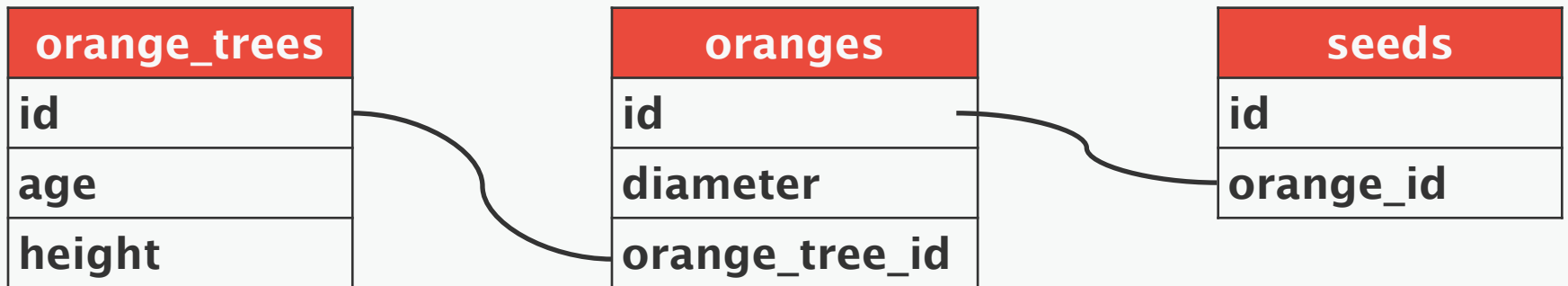
# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
end
```

```
class Orange < AR::Base
  belongs_to :tree,
    class_name: "OrangeTree",
    foreign_key: :orange_tree_id
end
```



# Spot the Errors

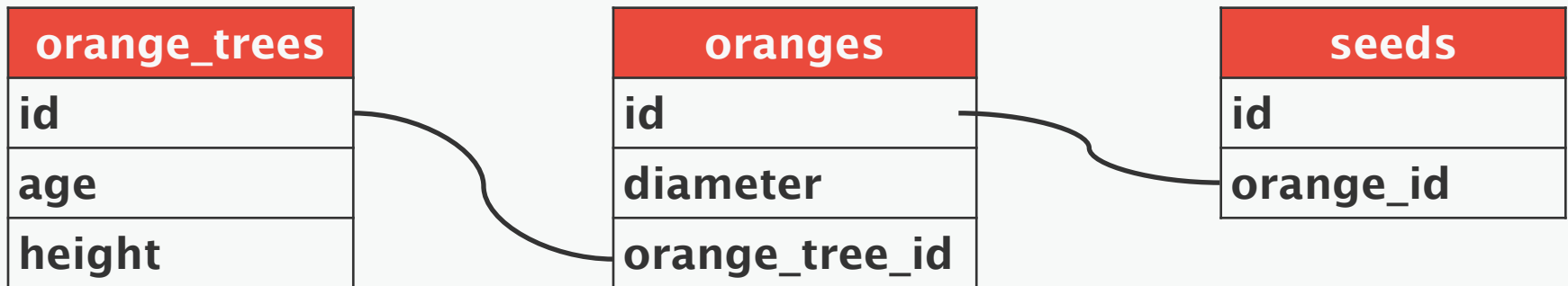


# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
  has_many :seeds,
    through: :oranges
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree
  has_many :seedlings
end
```

```
class Seed < AR::Base
  belongs_to :orange; end
```

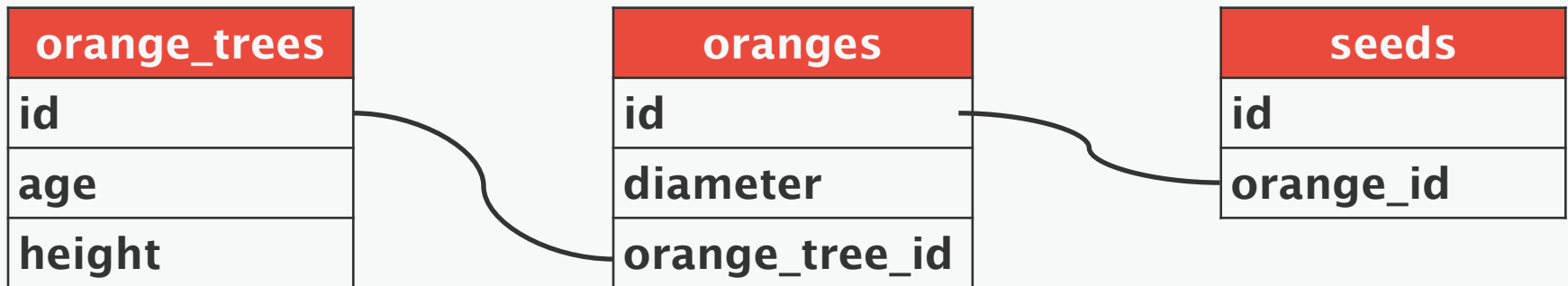


# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
  has_many :seeds,
    through: :oranges
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree
  has_many :seedlings
end
```

```
class Seed < AR::Base
  belongs_to :orange; end
```

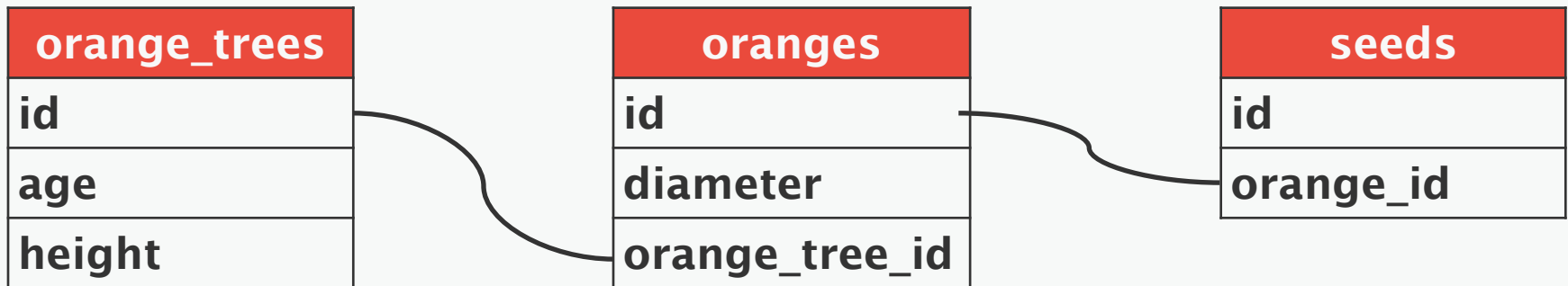


# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
  has_many :seeds,
    through: :oranges
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree
  has_many :seedlings,
    class_name: "Seed"
end
```

```
class Seed < AR::Base
  belongs_to :orange; end
```

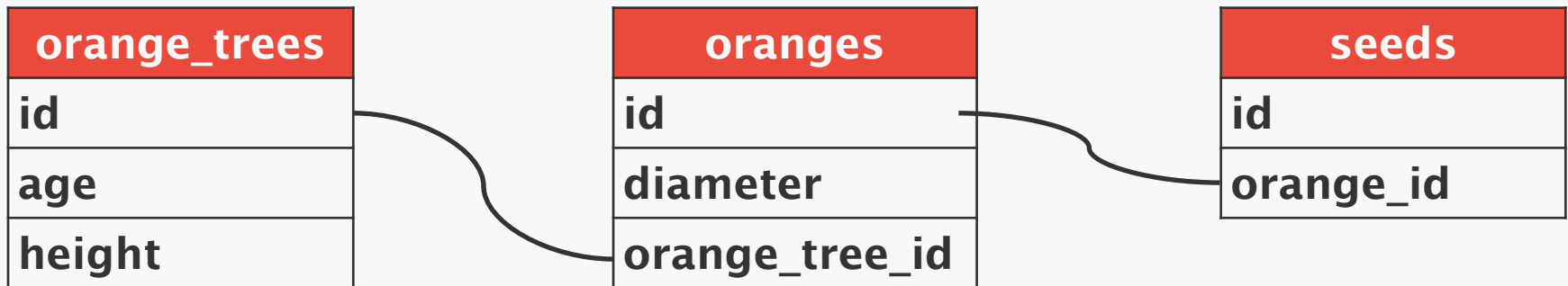


# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
  has_many :seeds,
    through: :oranges
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree
  has_many :seedlings,
    class_name: "Seed"
end
```

```
class Seed < AR::Base
  belongs_to :orange; end
```





# Spot the Errors

```
class OrangeTree < AR::Base
  has_many :oranges
  has_many :seeds,
    through: :oranges,
    source: :seedlings
end
```

```
class Orange < AR::Base
  belongs_to :orange_tree
  has_many :seedlings,
    class_name: "Seed"
end
```

```
class Seed < AR::Base
  belongs_to :orange; end
```

