

# **RELATIONSHIP GOALS**

# 3 TYPES OF DATABASE RELATIONSHIPS

MANY TO MANY

MANY TO ONE

ONE TO ONE

# 6 TYPES IN OF RAILS ASSOCIATIONS

# BELONGS\_TO

one to many relationship

```
class Author < ApplicationRecord
end

class Books < AppliicationRecord
  belongs_to :author
end
```

# HAS\_MANY

Many to one

```
class Author < ApplicationRecord
  has_many :books
end
```

```
class Books < ApplicationRecord
end
```

# HAS\_ONE

```
class Trainer < ApplicationRecord
  has_one :favorite, class_name: "Pokemon"
end
```

```
class Pokemon < ApplicationRecord
  belongs_to :trainer, as: :favorite
end
```

# HAS\_ONE\_THROUGH

```
class Supplier < ApplicationRecord
  has_one :account
  has_one :account_history, through: :account
end
```

```
class Account < ApplicationRecord
  belongs_to :supplier
  has_one :account_history
end
```

```
class AccountHistory < ApplicationRecord
  belongs_to :account
end
```



**THE MANY**

**MANY TO MANY**

# HAS\_MANY\_THROUGH

Join table has other attributes from foreign keys

```
class Trainers < ApplicationRecord
  has_many :teams
  has_many :pokemon, through: :teams
end

class Teams < ApplicationRecord
  belongs_to :trainer
  has_many :pokemon
end

class Pokemon < ApplicationRecord
  has_many :teams
  has_many :trainers, through: :teams
end
```

# has\_many\_and\_belongs\_to

Requires only join table w/o extra attributes

```
class Assembly < ApplicationRecord
  has_and_belongs_to_many :parts
end

class Parts < ApplicationRecord
  has_and_belongs_to_many :assemblies
end
```

# POLYMORPHIC: TRUE

```
class Pictures < ApplicationRecord
  belongs_to :imageable, polymorphic: true
end
```

```
class User < ApplicationRecord
  has_many :pictures
end
```

```
class Product < ApplicationRecord
  has_many :pictures
end
```

# INVERSE\_OF

```
class Author < ApplicationRecord
  has_many :books
end
```

```
class Book < ApplicationRecord
  belongs_to :book
end
```

# BI-DIRECTIONAL RELATIONSHIPS

```
# Without
a = Author.first
b = a.books.first
a.first_name == b.writer.first_name # => true
a.first_name = 'David'
a.first_name == b.writer.first_name # => false
# With
a = Author.first
b = a.books.first
a.first_name == b.writer.first_name # => true
a.first_name = 'David'
a.first_name == b.writer.first_name # => true
```

**END**