

ASSOCIATIONS

In Rails, an association is a relationship between two Active Record models.

They make common operations simpler.

Without associations:

```
genre = Genre.find_by(name: "Hip Hop/Rap")  
Track.where(genre: genre)
```

With associations:

```
genre = Genre.find_by(name: "Hip Hop/Rap")  
genre.tracks
```

FOREIGN KEYS

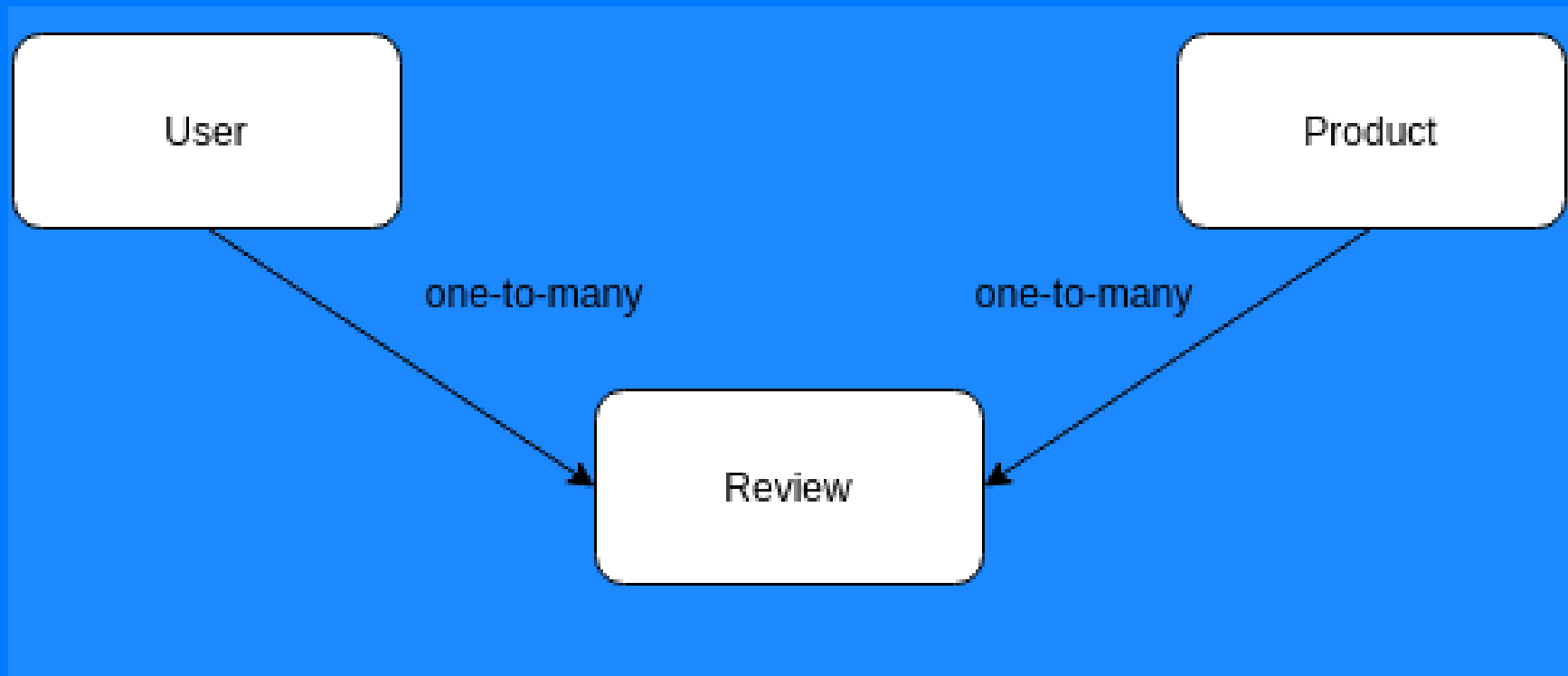
products

id	name
1	hat
2	jacket
3	boots
...	...
...	...

reviews

id	product_id	comment
1	1	Great!
2	3	Leaky :(
3	2	Super stylish.
4	3	Funky fresh!
...

ONE-TO-MANY



ONE-TO-MANY

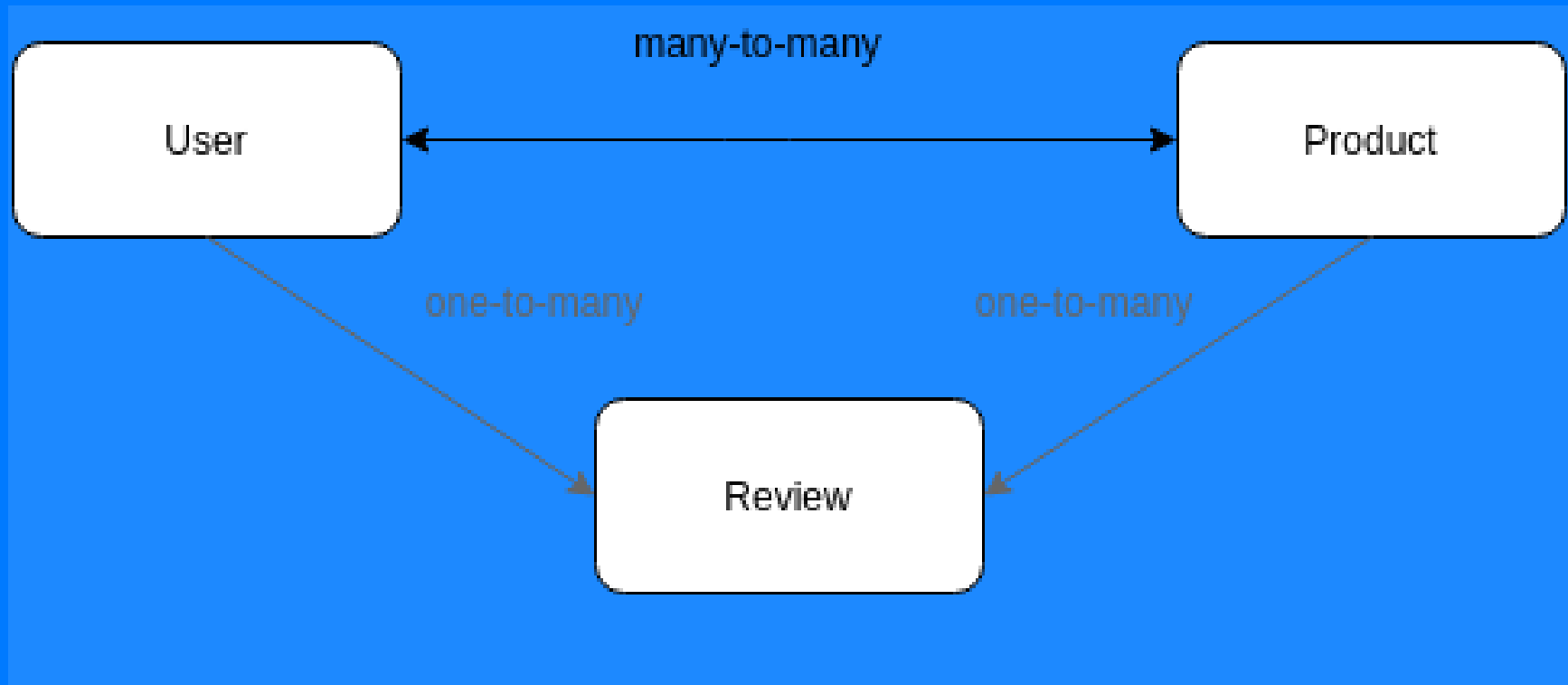
products

id	name
1	hat
2	jacket
1	boots
...	...
...	...

reviews

id	product_id	comment
1	1	Great!
2	3	Leaky :(
3	2	Super stylish.
4	3	Funky fresh!
...

MANY-TO-MANY



MANY-TO-MANY

users

id	name
1	Mina
2	Fabio
3	Natalie
...	...
...	...

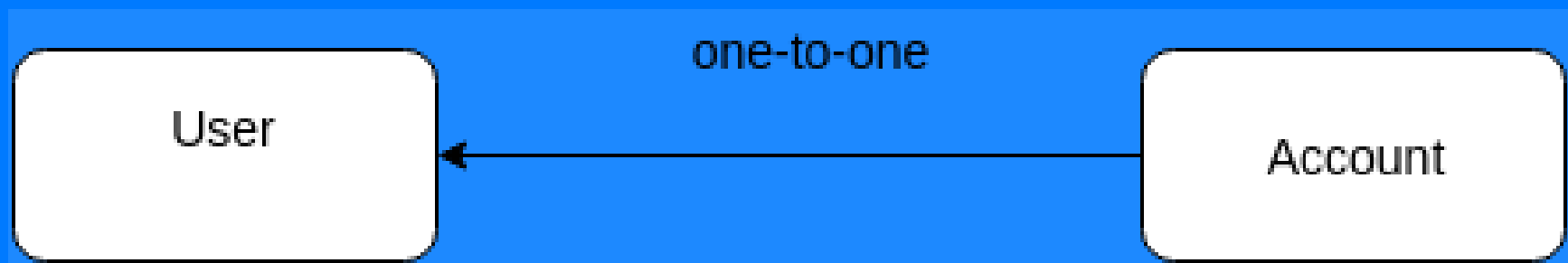
reviews

id	product_id	user_id	comment
1	1	2	Great!
2	3	2	Leaky :(
3	2	3	Super stylish.
4	3	1	Funky fresh!
...

products

id	name
1	hat
2	jacket
3	boots
...	...
...	...

ONE-TO-ONE



ONE-TO-ONE

users

id	name	email
1	Sandi Metz	smetz@gmail.com
2	_why	whytheluckystiff@gmail.com
3	Alan Turing	aturing@gmail.com

accounts

id	user_id	account_number
1	2	23432598
4	1	43853493
10	3	00483583

Associations **inform** our Rails models about the **existing structure** of our database.

Adding associations to our models **does not change** the database.

MODELS

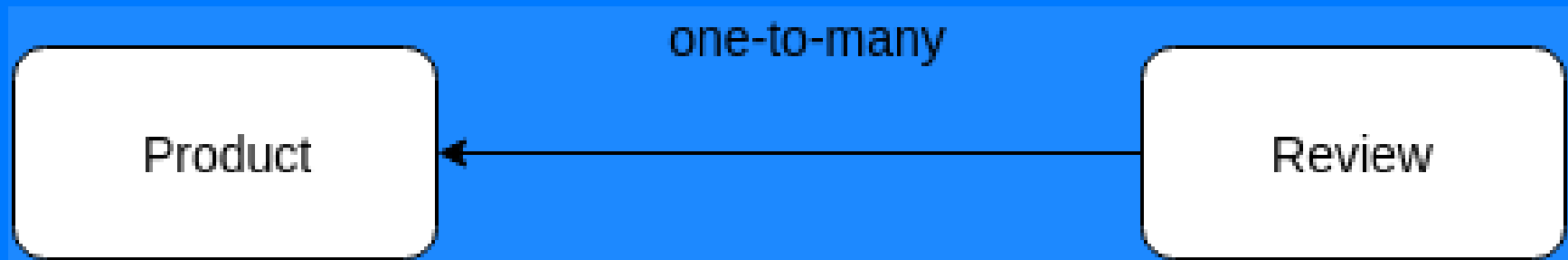
associations go here



DATABASE

foreign keys must exist here

ONE-TO-MANY



```
class Product < ActiveRecord::Base  
  has_many :reviews  
end
```

```
class Review < ActiveRecord::Base  
  belongs_to :product  
end
```

ONE-TO-MANY

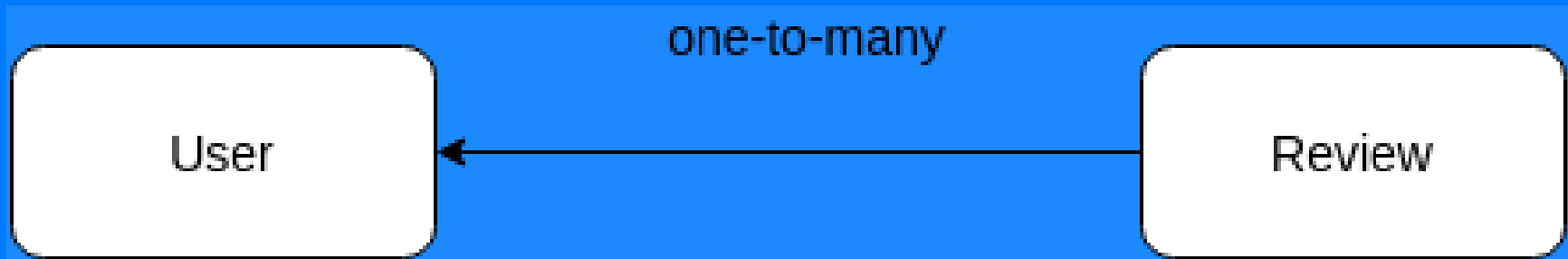
```
class Product < ActiveRecord::Base  
  
  has_many :reviews  
  
end
```

```
class Review < ActiveRecord::Base  
  
  belongs_to :product  
  
end
```

reviews

id	product_id	user_id	comment
1	211	1051	I love it!
2	99	47	I hate it!
3	99	80	I hate it too!

ONE-TO-MANY



```
class User < ActiveRecord::Base  
  
  has_many :reviews  
  
end
```

```
class Review < ActiveRecord::Base  
  
  belongs_to :product  
  belongs_to :user  
  
end
```


ONE-TO-MANY

```
class User < ActiveRecord::Base

  has_many :reviews

end
```

```
class Review < ActiveRecord::Base

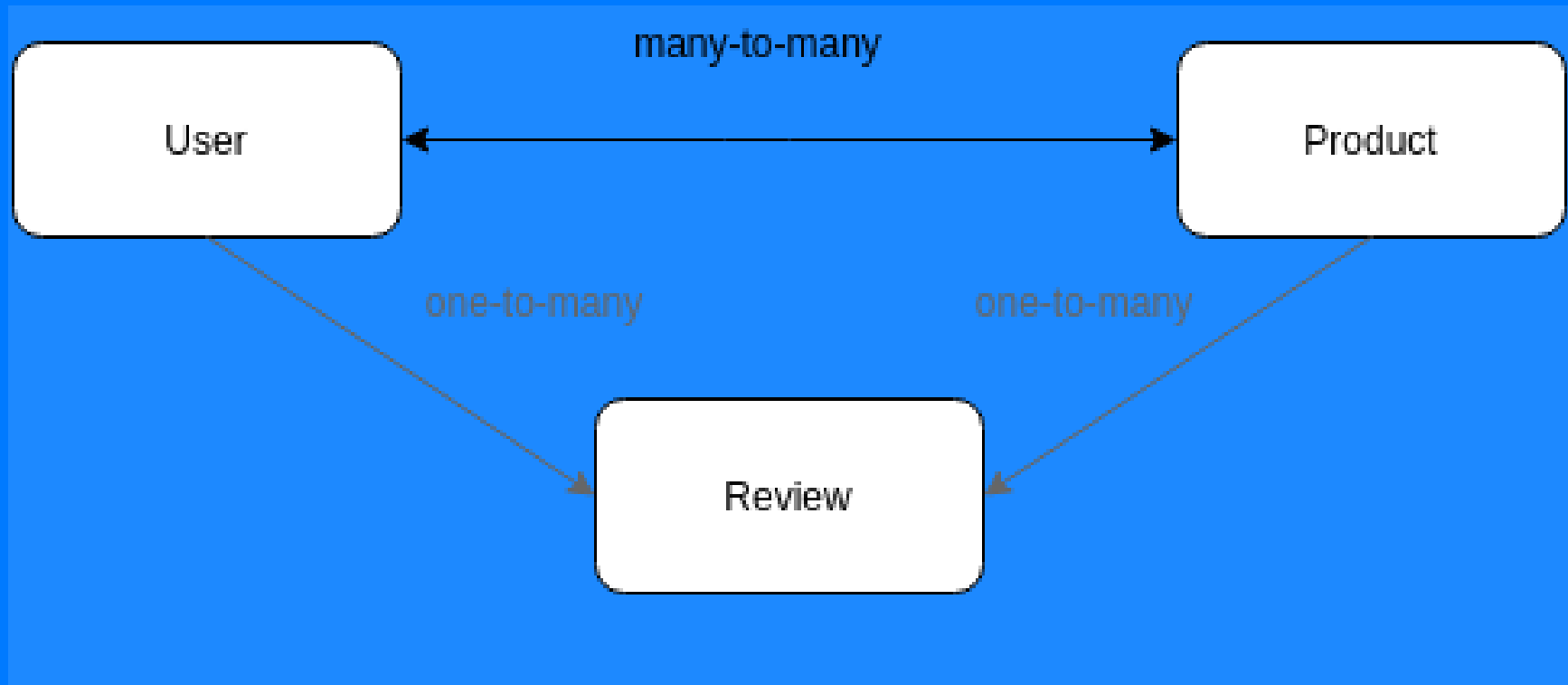
  belongs_to :product
  belongs_to :user

end
```

reviews

id	product_id	user_id	comment
1	211	1051	I love it!
2	99	47	I hate it!
4	100	47	Excellent quality.

MANY-TO-MANY



MANY-TO-MANY

```
class Product < ActiveRecord::Base

  has_many :reviews

end
```

```
class Review < ActiveRecord::Base

  belongs_to :product
  belongs_to :user

end
```

```
class User < ActiveRecord::Base

  has_many :reviews
  has_many :products, through: :reviews

end
```

```
has_many :products, through: :reviews
```

1. Find reviews
2. Ask each review for its product

MANY-TO-MANY

```
class Product < ActiveRecord::Base

  has_many :reviews
  has_many :users, through: :reviews

end

class Review < ActiveRecord::Base

  belongs_to :product
  belongs_to :user

end

class User < ActiveRecord::Base

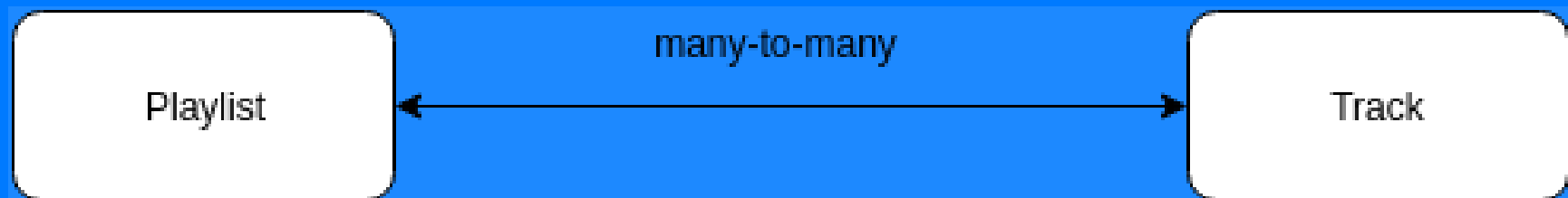
  has_many :reviews
  has_many :products, through: :reviews

end
```

```
has_many :users, through: :reviews
```

1. Find reviews
2. Ask each review for its user

MANY-TO-MANY



tracks

id	name	genre_id	unit_price	etc...
1	Moonlight Sonata	6	0.99	...
2	London Calling	2	0.99	...
3	Formation	1	0.99	...
4	Where Does the Good Go	3	1.99	...
5	You're My Waterloo	2	0.99	...

playlists

id	name
1	Summer jams
2	Songs for crying
3	Work playlist
4	Favourite songs
5	Sappy mixtape

playlists_tracks

playlist_id	track_id
1	1
1	2
2	1
2	5


```
class Playlist < ActiveRecord::Base

  has_and_belongs_to_many :tracks

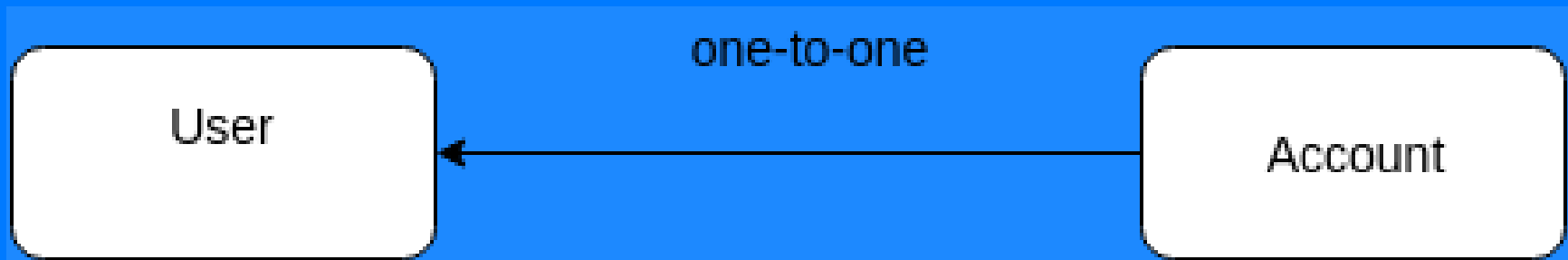
end

class Track < ActiveRecord::Base

  has_and_belongs_to_many :playlists

end
```

ONE-TO-ONE



```
class User < ActiveRecord::Base  
  has_one :account  
end
```

```
class Account < ActiveRecord::Base  
  belongs_to :user  
end
```

ONE-TO-ONE

```
class User < ActiveRecord::Base  
  has_one :account  
end
```

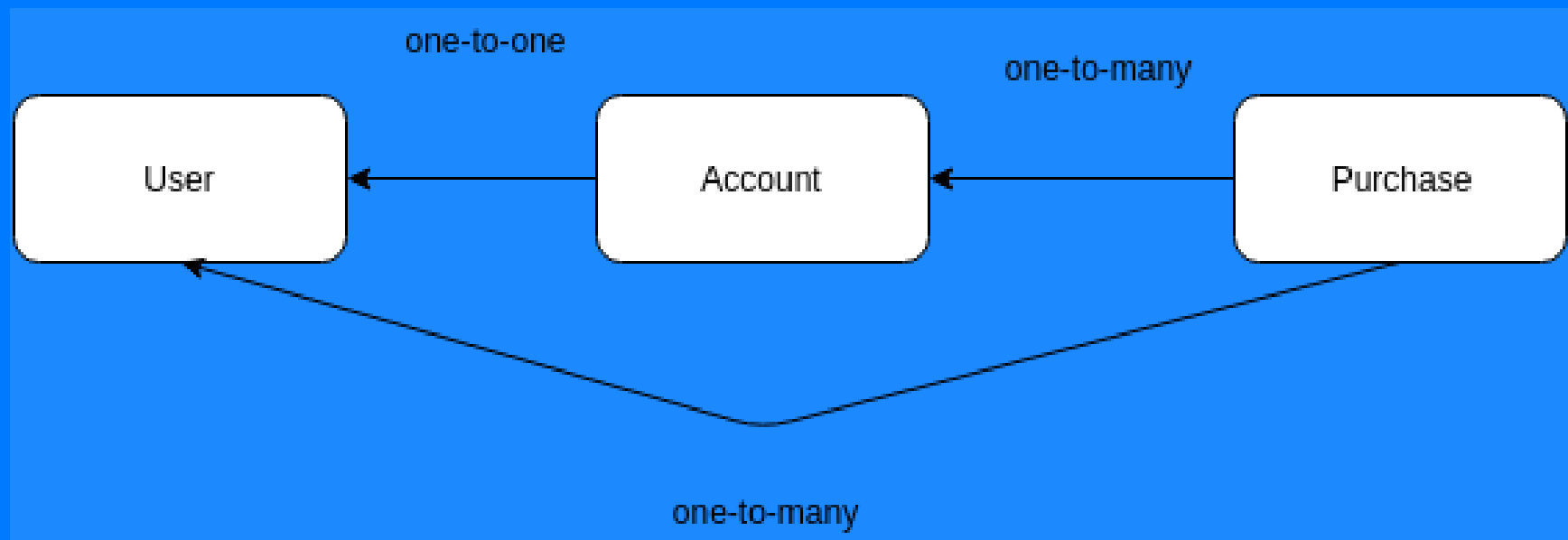
```
class Account < ActiveRecord::Base  
  belongs_to :user  
end
```

accounts

id	user_id	account_number
1	2	23432598
4	1	43853493
10	3	00483583

ONE-TO-MANY

has_one through



users

id	name	email
1	Sandi Metz	smetz@gmail.com
2	_why	whytheluckystiff@gmail.com
3	Alan Turing	aturing@gmail.com

accounts

id	user_id	account_number
1	2	23432598
4	1	43853493
10	3	00483583

purchases

id	account_id	amount	etc...
9	4	25.50	...
10	4	250.00	...
11	10	10.99	...

has_one through

```
class Purchase < ActiveRecord::Base

  belongs_to :account
  has_one :user, through: :account

end

class Account < ActiveRecord::Base

  belongs_to :user
  ...

end

class User < ActiveRecord::Base

  ...

end
```

~~belongs_to :user, through: :account~~

has_one :user, through: :account