



# Kurs programowania Ruby on Rails

---

## Asocjacje

# Asocjacje



- Asocjacje definiują połączenie między modelami.
- Ułatwiają one życie programistom:)

# Bez asocjacji...



```
class Organization < ActiveRecord::Base  
end
```

```
class Person < ActiveRecord::Base  
end
```

```
@person = Person.create(:name => 'Michał',  
:organization_id => @organization.id)
```

```
@people = Person.find_by_organization_id(@organization.id)
```

```
@people.each do |person|  
  person.destroy  
end
```

# Tworzenie asocjacja Rails



```
@organization.people.create(name: 'Michał')
```

```
@organization = Organization.first
```

```
Person.create(name: 'Piotr', surname:  
'Kowalski', organization: @organization)
```

# Proces tworzenia asocjacji



- musimy stworzyć klucz obcy poprzez niego odwołujemy się do powiązanej tabeli – w naszym przypadku musimy utworzyć np. Dla Modelu Employee organization\_id odwołujące się do Organization
- dodatkowo musimy dodać do modelu typ asocjacji np. belongs\_to

# References



- tworzymy klucz obcy poprzez niego odwołujemy się do powiązanej tabeli – w naszym przypadku musimy utworzyć np. Dla Modelu Employee organization\_id odwołujące się do Organization
- dodatkowo tworzy nam zdefiniowaną relacje „w jedną stronę”

# References - Rails



Rails g model Book title:string description:text isbn:string  
author:references

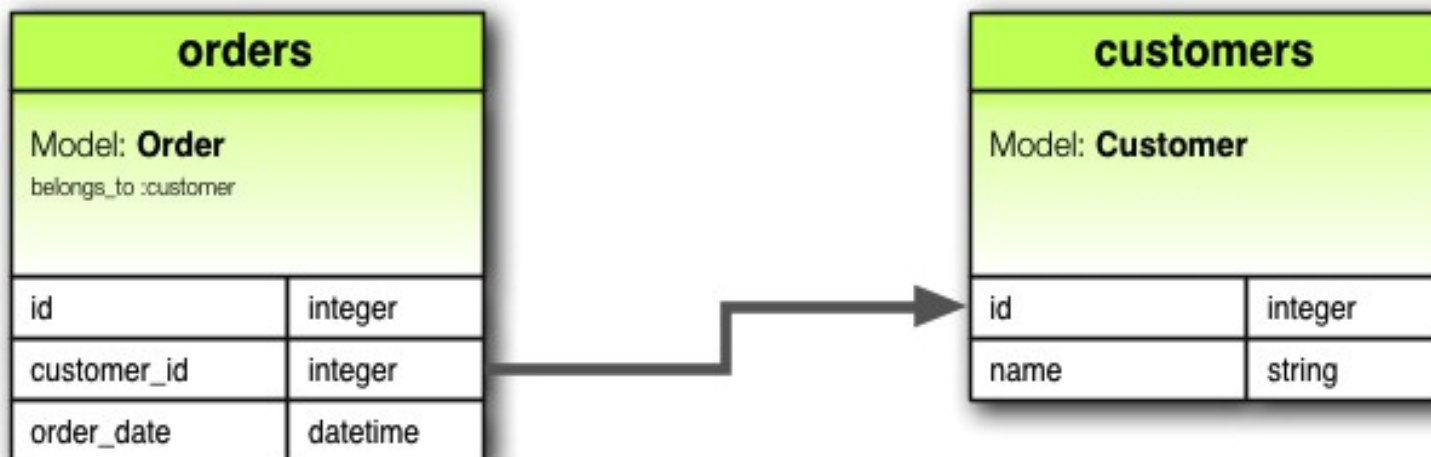
# Migracja



```
class CreateBooks < ActiveRecord::Migration
  def change
    create_table :books do |t|
      t.string :title
      t.text :description
      t.string :isbn
      t.references :author, index: true #tworzy nam author_id oraz
index w bazie danych
      t.timestamps
    end
  end
end
```



# Asocjacja belongs\_to



```
class Order < ActiveRecord::Base
  belongs_to :customer
end
```

# belongs\_to opcja class\_name



-Standardowo jeśli piszemy belongs\_to :author asocjacja odwołuje się do Klasy Author , natomiast jeśli chcemy zaznaczyć ,że klasa ma odwoływać się np. do Person piszemy :class\_name =>'Person', a asocjacja ma się nazywać author

```
class Book < ActiveRecord::Base
  belongs_to :author, :class_name =>"Person"
end
```



# Kurs programowania Ruby on Rails

---

Asocjacje

Koniec