NBU – Note for better understanding which is added from me

Associations in Rails

* When we used the scaffold generator for the model like posts and comments in our blog application, the Active Record design pattern is “pre-wired”.
* We did make the connection between posts and comments in the database – recall that a post\_id can be stored with each comment
* To make our models in Rails fully functional we need to add associations – each post needs to know the list of comments associated with it, and each comment needs to know which post it belongs to.
* There’s a many-to-one relationship between comments and posts – a post has many comments, and comment belongs to a post:



* The ActiveRecord module contains a set of class methods for tying objects together through foreign keys.
* To enable these, We must declare associations within our models using the following:

|  |  |  |
| --- | --- | --- |
| Relationship | Model with no foreign key | Model with foreign key |
| One-to-one | has-one | belongs\_to |
| many-to-one | has\_many | belongs\_to |
| many-to-many | has\_and\_belongs\_to\_many | \* |

\*The foreign keys for each model are stored in a join table.

- NBU

- Model with on foreign key🡪Parent Table

- Model with foreign key 🡪Child Table

- Add decoration in our blog application

- In Post model

Class Post < ActiveRecord::Base

has\_many :comments

end

In Comment model

Class Comment < ActiveRecord::Base

belongs\_to :post

end

* Since comments are dependent on post then if we want to delete all comments when delete a post write following code as like:

Class Post < ActiveRecord::Base

has\_many :comments, dependent::destroy

end