Many Subclass inherits

User   Products   Order
has many   has many   belongs to
order   orders   as orderable
as orderable   polymorphic

* Single Table Inheritance | Polymorphic assoc

Single Table Inheritance	Polymorphic assoc
• Many subclass inherit from one superclass with all data in the same table in db.	• One model belong to several other models using single association.
• The superclass has a "(type)" column to determine, which sub-class an obj belongs to.	Each model include polymorphic model, has its own table in db. ex (type find column)

ex Vehicle 3 Superclass

↓ Iske andr

Same Attribute Should be stored among all subclass.

(Color, price, purchased)

↓

(Cars) (bike) (Bicycle)

ex - Fb data

↙ ↘
(groups) (Individual)
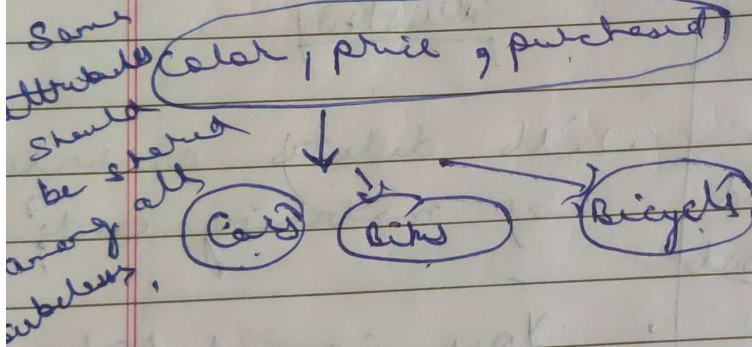
↘
They both have diff data.
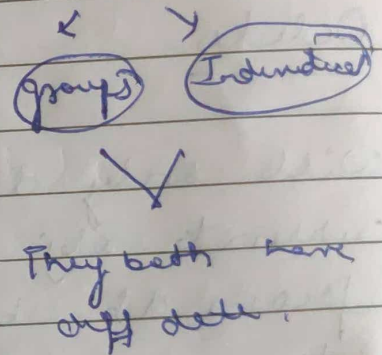
• One table for all classes

• Shared attributes

• Uses table per class.

ex - User   Product   Order

has many order as orderable.

has many order as orderably

belongs to order polymorphic tr

orderable - id
orderable - d

* Lambdas {} | Procs   Proc.new {}

Lambdas {}	Procs Proc.new {}
• Checks the no. of arguments.	• Doesn't care if wrong no. of argument passed

ex. lam = lambda {|x| puts x} proc = Proc. new { |x| puts x

• lam. call (2); prints 2 • proc. call (2); prints 2

• lam. call ; errors   proc. call ; Returns nil
  as no. argument
  passed.

• lam. call (1, 2, 3);   • proc. call (1, 2, 3);
  Argument error: prints out 1 &
  wrong no. of arg.   forgets abt other
  {} for 1 y   elements.

* Yield →

Is a ruby keyword that calls a
block when you see it.
When you use yield keyword, the
code inside block will run.

ex - def print
        yield
     end
print {{ puts " Block is being run"}.

ex def print _ twice
       yield   o/p hello
       yield   hello.
     end
     _____ twice { puts "hello"}

self → Access instance of a class

Saathi

Difference b/w lambda & proc →

ex → def my_method
　　　puts "before proc' Ist executed
　　　　　　　Lambda
　　　　　　　or
　　my_proc = proc {　　　IInd
　　　　↑ puts "Inside proc"
　　　　　return
　　　　　Iski wajah se phele print hua
　　}
　　my_proc.call　　　Ird
　　　puts "after proc'
end.

Calling → my_method

O	p → Before proc	when proc & lambda
Inside proc	are used w/o break &
after proc.	return statement

O	p → before proc	when proc used with
Inside proc	return so it will come
	outs from proc & method
	too.

In case of lambda if we are giving
return statement then it will throw
us out of proc but not with the
method.

| O|p → before proc |
| Inside |
| after |

Assigning a controller & an action to a URL.

app/config/route

* <u>Routes</u> - way to redirect incoming req. to controller & act

- Simple match route
- Default mat route
- root route
- resourceful route

(Rest)

Get - Retreive items from resource
use in (link)

Post - Create new items in 1
(form)

Patch - Update existing item
(form)

Delete - Delete " " "
(forms, links)

Put.

* <u>View templates</u> →

erb → embedded ruby

(@)
Instance variable used in view.

* <u>Links</u> →

< a href = "/demo/index" >
        route

Controller name should be plural
_____ etc.

* Restful routes →
Provides a mapping b/w HTTP, new
controller actions and CRUD oper" in
db.

* Rest HTTP verb →

Action	HTTP	Description
index	Get	Show all items
show	"	Show item with id
new	"	Show new form
create	post	Create an item
edit	get	Show edit form for item id
update	patch	Update
delete	get	Show delete form with id
destroy	delete	" item with id

* Resourceful routes → [config / routes.rb

ex → resources : subjects
     resources : pages

(< ·/· =   ·/·>)