**Models and Tables**

**users**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id (integer)  primary key | name  (string) | email  (string) | password  (string) | phone  (integer) | contact\_method (  0 – email   1. Phone 2. Text) | role (integer)  **(cannot be nil)**  0  1  2  3 |

**realtor**

|  |  |  |
| --- | --- | --- |
| id (integer)  primary Key | user\_id (integer)  (foreign key for Table users) | company\_id (integer)  (foreign key for Table companies) |

**hunters**

|  |  |
| --- | --- |
| id (integer)  primary key | user\_id (integer)  (foreign key for Table users |

**b**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id  (Integer)  primary Key | location  (string) | square\_ft  (Float)  (in ft2) | year\_built  (Integer) | style  (0 – single  family  home   1. Apartment 2. Condo) | price  (float) | floor  (integer) | basement  (Char(Yes/No) | owner (string) | contact\_info  (string) | realtor\_id  (foreign key for Table realtors) | company\_id  (Integer)  Foreign key for Table companies | pic\_linsk  (string)  (the address to the folder of the pics) |

**houses**

**companies**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id (integer)  Primary Key | name (string) | website (string) | address  (string) | **size**  **(string)**  **format:**  **nil** or  **30-50** or  **30** or  **30+** | year\_founded  (integer) | **revenue**  **(string)**  **format:**  **nil** or  **30-50** or  **30** or  **50+** | synopsis (text) |

**inquiries**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id(integer)  Primary Key | hunter\_id (integer)  foreign key from users | house\_id (integer)  foreign key from  houses | subject (string) | message (string) | response (string) |

**interests**

|  |  |  |
| --- | --- | --- |
| id(integer)  Primary Key | hunter\_id (integer)  foreign key from users | house\_id (integer)  foreign key from  houses |

**Administrator Info:**

name: "admin",

email: "admin@ncsu.edu",

password: "oodesign",

phone: 919517517,

contact\_method: 0,

role: 1,

curr\_role: 0,

**Association between models:**

**User**

has\_one :realtor, dependent: :destroy

has\_one :hunter, dependent: :destroy

**Realtor**

has\_many :houses

belongs\_to :user

belongs\_to :company

**House**

has\_many :inquiries, dependent: :destroy

has\_many :interests, dependent: :destroy

has\_many :hunters, :through => :inquiries

has\_many :hunters, :through => :inquiries

belongs\_to :realtor

belongs\_to :company

**Company**

has\_many :realtors

has\_many: houses, dependent: :destroy

**Inquiry**

belongs\_to :hunter

belongs\_to :house

**Interest**

belongs\_to :hunter

belongs\_to :house

**Hunter**

has\_many :inquiries, dependent: :destroy

has\_many :interests

has\_many :houses, :through => :inquiries

belongs\_to :user

**Modifications I have done to comply with Rails convention.**

1. Change ***numeric*** type to either ***integer*** type or ***float*** type.
2. Change “Table *companies*‘s *synopsis* field to “text” as it is long text.
3. Conventionally in Rails, foreign key is formatted as “*name*\_id”, where *name* originates from Table *names*.
4. use simple names for models and tables
5. Use singular names as Model names and plural names as table names. Specifically, see below.

irb(main):007:0> 'company'.pluralize

=> "companies"

irb(main):008:0> 'inquiry'.pluralize

=> "inquiries"

**Example CURD operations and association references :**

**We can go to project root folder, type ‘rails console’, then use the models to play with the database, and test the association reference operations between models.**

**A:To create our administrator named “Prachi Gupta”**

admin = User.create(name: " Prachi Gupta", email: " pgupta24@ncsu.edu", password: "oodesign", phone:517517517, ct\_method: 0, role: 1, curr\_role: 5)

#Note: This only needs to be done once. I have already done it.

**B:To create a hunter named “Hasham Mukhtar”, be careful not to set role as 1 (admin)**

#First method (recommended) - create user\_hasham first, then use user\_hasham.create\_hunter to create its hunter role, auto-binding

user\_hasham = User.create(name: " Hasham Mukhtar", email: " hmukhta@ncsu.edu", password: "oodesign", phone: 517517517, ct\_method: 1, role: 4, curr\_role: 5)

hunter\_hasham = hasham.create\_hunter

#Second method (not recommended) - create user\_Hasham and hunter\_Hasham separately, then bind them together mannually

user\_hasham = User.create(name: " Hasham Mukhtar", email: " hmukhta@ncsu.edu", password: "oodesign", phone: 517517517, ct\_method: 1, role: 4, curr\_role: 5)

hunter\_hasham = Hunter.create

user\_Hasham.hunter = hunter\_Hasham

**C: To create realtor named “Hancheng Wu”**

#realtor has user\_id and company\_id as foreign keys. Thus, we must obtain company\_id before we can call ‘user.create\_realtor’.

#create company ‘’NCSU” first

ncsu = Company.create(name: "NCSU", website: "www.ncsu.edu", address: "Raleigh, NC", low\_bd: 20000, up\_bd: 0, year\_founded: 1887, revenue: 100000000, synopsis: "NCSU is a research university")

#then create user ‘Hancheng Wu’

user\_henry = User.create(name: "Hancheng Wu", email: " hwu16@ncsu.edu", password: "oodesign", phone: 888888888, ct\_method: 1, role: 2, curr\_role: 5)

#create realtor ‘Hancheng Wu’, here we need to pass company\_id because it is a foreign key

realtor\_henry = user\_henry.create\_realtor(company\_id: ncsu.id)

#alternatively, we can create realtor ‘Hancheng Wu’ by ncsu.create\_realtor as well

realtor\_henry = ncsu.create\_realtor(user\_id: user\_henry.id)

**D: How to Read, Update tables.**

admin = User.find\_by\_role(1)

admin.password = "new\_password"

admin.save

output:

irb(main):028:0> admin = User.find\_by\_role(1)

User Load (0.3ms) SELECT "users".\* FROM "users" WHERE "users"."role" = ? LIMIT ? [["role", 1], ["LIMIT", 1]]

=> #<User id: 1, name: "Prachi Gupta", email: "pgupta24@ncsu.edu", password: "oodesign", phone: 919517517, ct\_method: 0, role: 1, curr\_role: 0, created\_at: "2018-09-25 02:55:24", updated\_at: "2018-09-25 04:59:27">

irb(main):029:0> admin.password = "new\_password"

=> "new\_password"

irb(main):030:0> admin.save

(0.2ms) begin transaction

SQL (0.7ms) UPDATE "users" SET "password" = ?, "updated\_at" = ? WHERE "users"."id" = ? [["password", "new\_password"], ["updated\_at", "2018-09-25 05:00:06.269468"], ["id", 1]]

(3.7ms) commit transaction

=> true

**E: How to Delete a record.**

#create a hunt called tmp

user\_tmp = User.create(email: "tem@ncsu.edu")

hunt\_tmp = user\_tmp.create\_hunter

Hunter.all

#delete user tmp, infact both user tmp and hunt tmp are deleted with the following one method,

user\_tmp.destroy

#verify if deletion is successful

User.all

Hunter.all

**More association reference operations can be found here:**

[**https://guides.rubyonrails.org/association\_basics.html#has-one-association-reference**](https://guides.rubyonrails.org/association_basics.html#has-one-association-reference)

**Table users**

**remove the curr\_role field,**

**change the validation of the role field,**

**Create new Table**

**interests**