

Web System Development with Ruby on Rails



Day 7(8/Nov/2012)

Relational Database

Today's Theme

- Learn Relation Structure in Relational Database
- Understand how to describe the relational structure
- Add new Code Table to the relational database

Modeling Data

- ❑ What does Data modeling mean?
 - ❑ Describe data structure
 - ❑ Relations between properties
 - ❑ Describe Constraint for Conformance
 - ❑ Conformance – Having no contradiction between components
 - ❑ Describe how to manipulate data
 - ❑ Separate input fields and calculated fields
- ❑ Define Schema by data modeling

Relation Type and Entity Type

- ❑ Relation : is a relation and/or a Table
 - ❑ There are two types of relations, both called "Relation" which means;
 - ❑ Entity, or
 - ❑ Relation

- ❑ Entity
 - ❑ Real world object (not physical sometimes)

- ❑ Relation
 - ❑ Relation between two entities

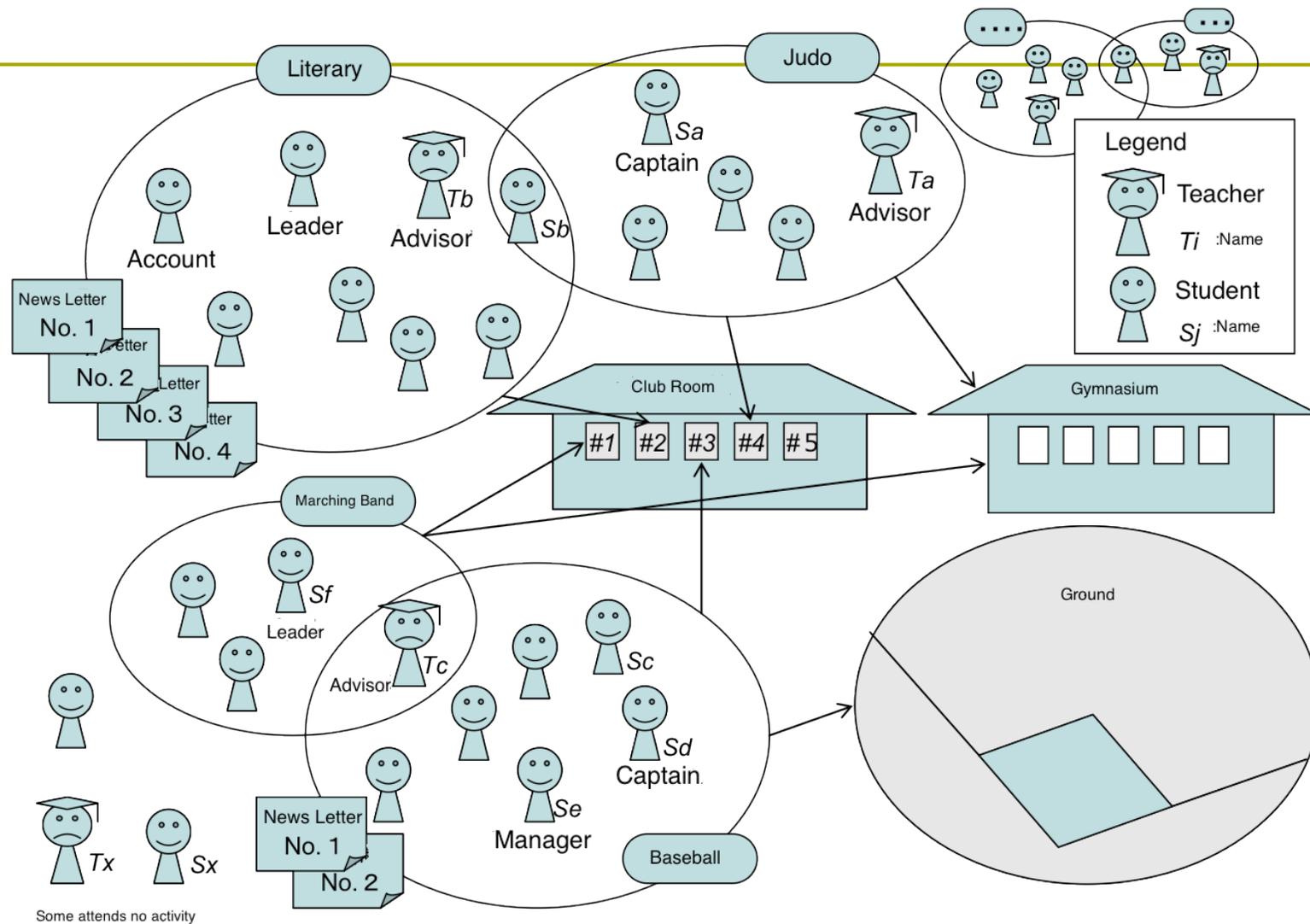
What is Entity

- Entity Type:
 - physically exists in the real world;
 - students, teachers, lecture rooms, pizza(menu)
 - is identical in the real world;
 - University, subjects, flights, club teams
 - They cannot be seen or even touched.
- Identical Entity
 - University Buildings physically exists, but the organization “university” does not necessarily require the buildings.

What is relation type

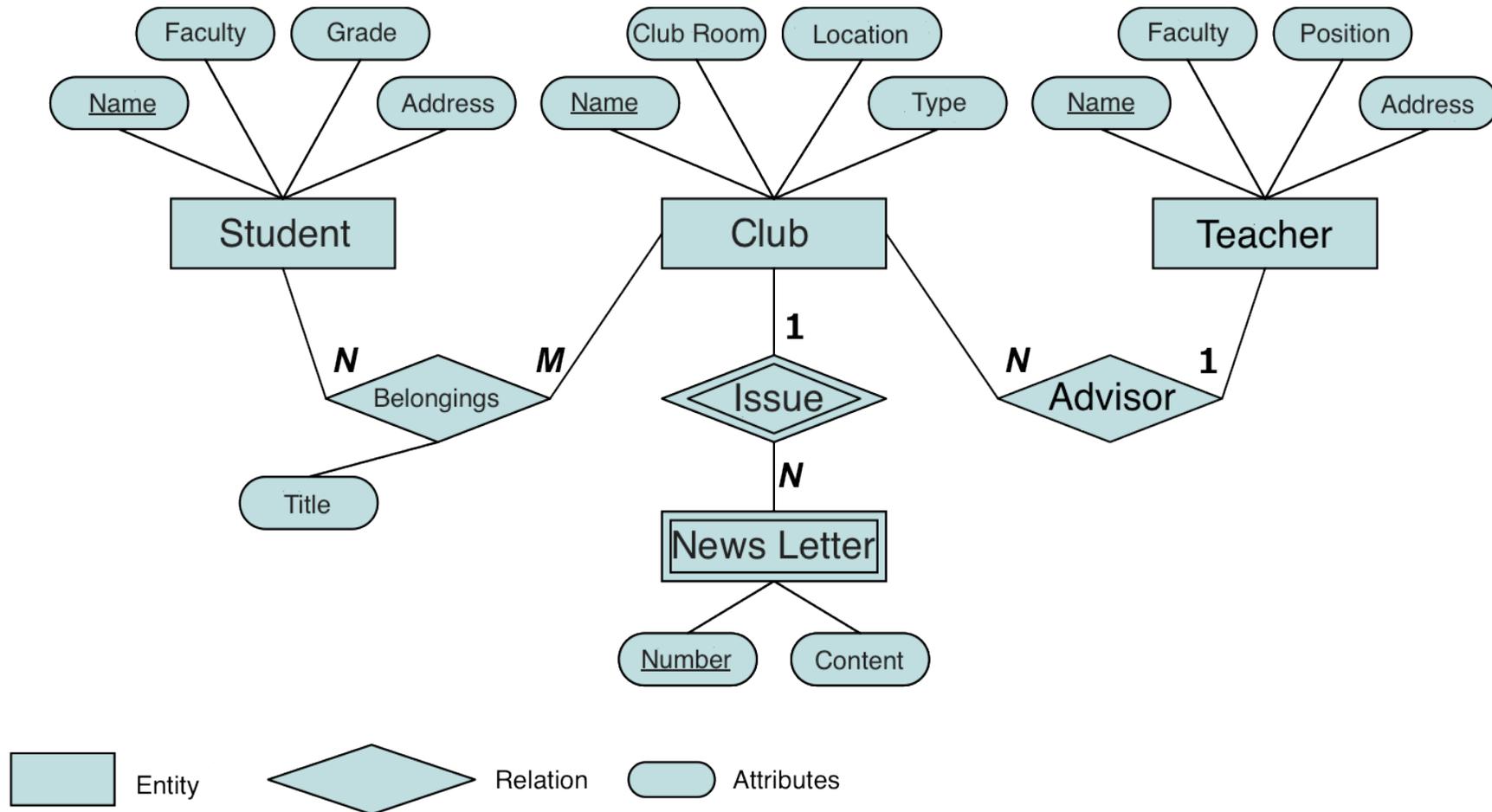
- Relation Type
 - Represents the relation between two entities
 - Enrollment: Students “Enroll” in subjects
 - Order: Customers “Order” pizza
 - Reservation: Customers “reserve” flights
 - Rental: Customers “rent” DVD
 - One customer can rent many DVDs
 - One customer can order many pizza
 - One student can enroll in many subjects

Example: “Club Activity”



Original Figure: Haruo Hayami, Practicing Base of Relational Database, 2008, Corona Publishing, in Japanese

ER figures of “Club Activities”



Original Figure: Haruo Hayami, Practicing Base of Relational Database, 2008, Corona Publishing, in Japanese

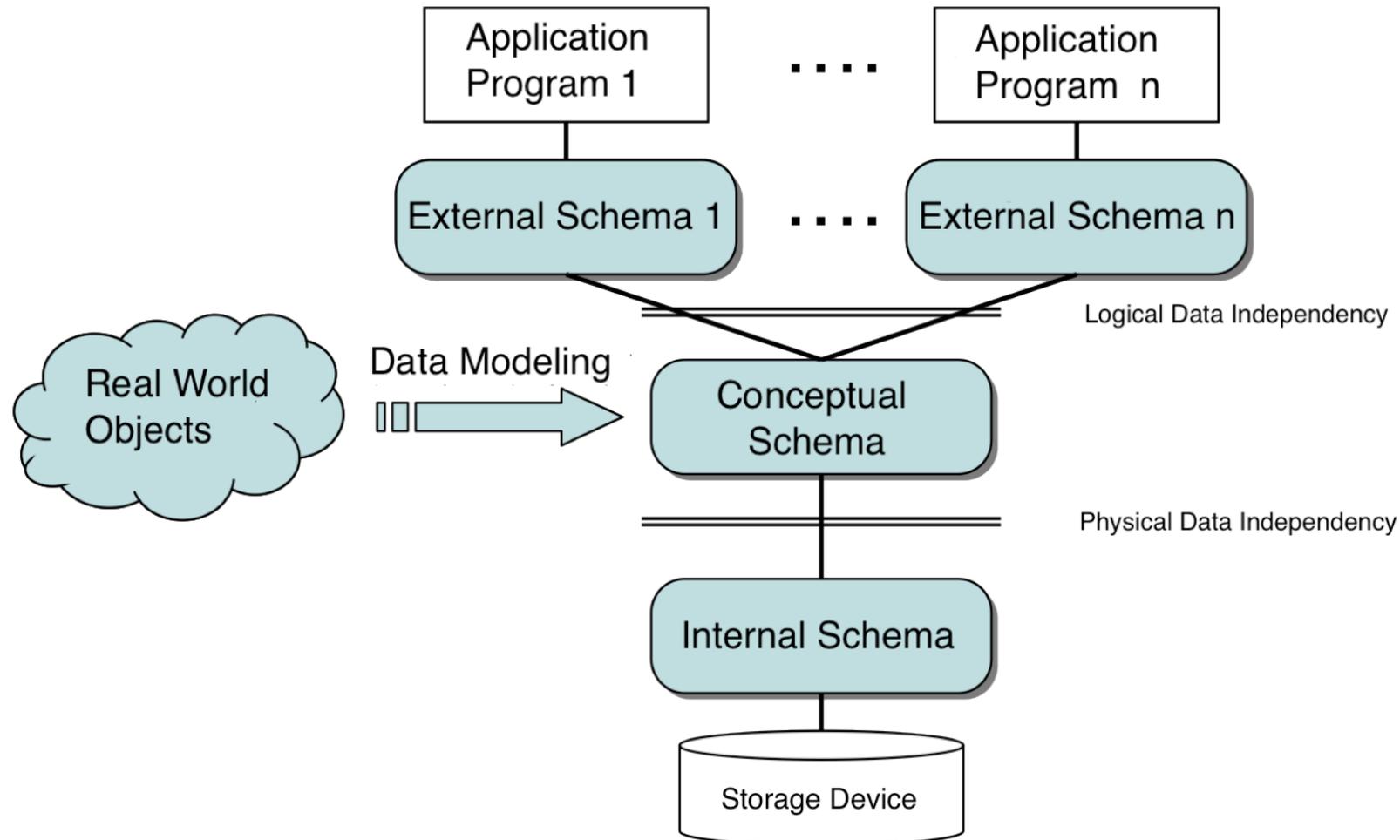
What is Schema

- ❑ Schema
 - ❑ Data structure obtained by describing application data model
 - ❑ Relations between components of data

- ❑ Three Layer Schema model:
 - ❑ External Schema, Conceptual Schema, and Internal Schema

ANSI/X3/SPARC

3 Layer Schema Architecture



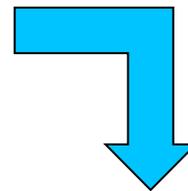
First Normal Form(1NF)

Relations contain **only "atomic" values**

Only when all attributes in the relation are "atomic values," the relation is First Normal Form.

Customer

Customer ID	First Name	Surname	Telephone Number
123	Robert	Ingram	555-861-2025
456	Jane	Wright	555-403-1659 555-776-4100
789	Maria	Fernandez	555-808-9633



Customer Name

Customer Telephone Number

<u>Customer ID</u>	First Name	Surname	<u>Customer ID</u>	<u>Telephone Number</u>
123	Robert	Ingram	123	555-861-2025
456	Jane	Wright	456	555-403-1659
789	Maria	Fernandez	456	555-776-4100
			789	555-808-9633

Original Figure: Wikipedia

Normalization

- ❑ First Normal Form (**1NF**)
 - ❑ Consist of only “**atomic**” values
- ❑ Second Normal Form (**2NF**)
 - ❑ Specifically: a table is in 2NF if and only if it is in 1NF and **no non prime attribute is dependent** on any proper subset of any candidate key of the table.
- ❑ Third Normal Form (**3NF**)
 - ❑ It is in 2NF and every non-prime attribute of R is **non-transitively dependent** (i.e. directly dependent) on every superkey of R.

Intuitive Guide for Normalization

- Mathematically difficult, but on intuition...
- Student A moved his apartment house;
 - He submitted the “notice of address movement,” **only one business section**, but all the other business section has come to know his “address movement.”
- In the normalized form, necessary change of information are transferred to all the necessary sections.

The principle of Normalization

- ❑ Means to maintain the data integrity
 - ❑ Data **should be shared and stored only in one place.**
 - ❑ Normalization is one of the mean to maintain the **integrity.**
- ❑ One fact in one place.
- ❑ One fact in **Only one** place.
 - ❑ This is the design principal of database.

Today's practice theme

- Make one table of category, then add relations from memos to categories.
- Step 1: add one field into memos
 - Add one migration file (to add and/or remove up relation field,) by generating migration file.
- Step 2: execute migration
- Step 3: add drop down list of categories in the memos table.

Scaffolding of Category

Prepare scaffolding and migration file to create categories table

(We have done this the last day.)

(See Page 38 of Day 6 Slides)

`rails generate scaffold categories name:string`

- Data input views, model, and controller are generated.
- Migration file of category is generated.

When completed then run

`rake db:migrate`

Migration to Add Column

We generate migration file of adding Column to Memos table by typing the following command;

```
rails generate migration AddCategoryIdToMemos
```

Note that the generated Migration file needs to be modified.

```
[root@cisnote memopad]# rails g migration AddCategoryIdToMemos
      create  db/migrate/20121106231235_add_category_id_to_memos.rb
```

Migration File

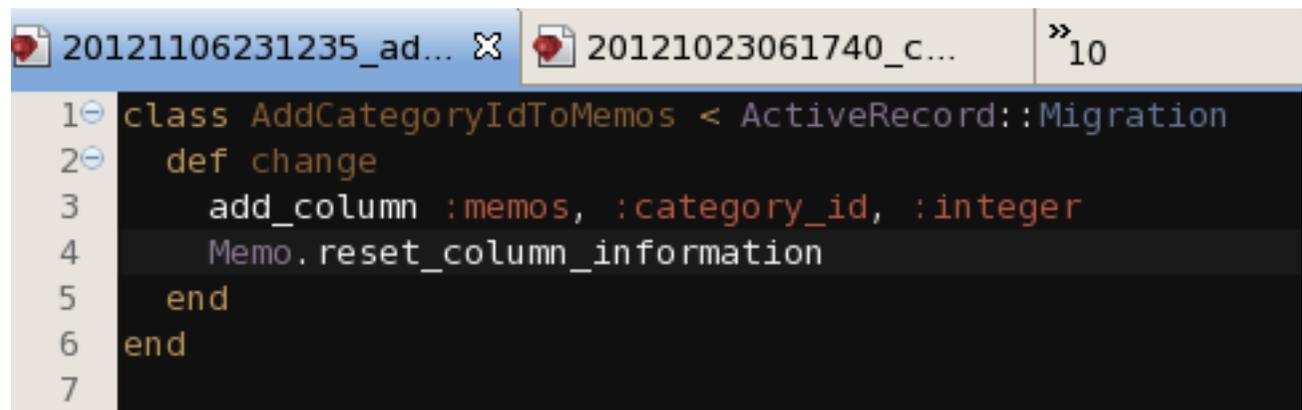
Add the following two lines in the

`2012mmddhhMMss_add_category_id_to_memos.rb` file;

`add_column :memos, :category_id, :integer`

`Memo.reset_column_information`

In `ActiveRecord::Migration`, there are 'add_column' and some other methods to change schema available.



```
20121106231235_ad... x 20121023061740_c... »10
1 class AddCategoryIdToMemos < ActiveRecord::Migration
2   def change
3     add_column :memos, :category_id, :integer
4     Memo.reset_column_information
5   end
6 end
7
```

Migration other than create/drop

- ❑ Migration allow us not only create/drop tables, but also add, rename, and change columns

See <http://guides.rubyonrails.org/migrations.html>

- ❑ Add column
 - ❑ `add_column :table, :column, :type, :options`
- ❑ Rename column
 - ❑ `rename_column :table, :column, :newColname`
- ❑ Change column type
 - ❑ `change_column :table, :column, :type, :options`
- ❑ Remove_column
 - ❑ `remove_column :table, :column`

Migration and the name

- ❑ Insert link_id to categories in Memos table;
 - ❑ category_id can be used as a link.
- ❑ There is a rule in Ruby

Convention over Configuration

- ❑ The name of Relation field is "name_id" where the name should be singular form.
 - ❑ It will be `category_id` in this case.

Migration of Adding Column

When you finished updating the migration file to add columns, then do the migration.

Type:

`rake db:migrate`

```
[root@cisnote memopad]# rake db:migrate
== AddCategoryIdToMemos: migrating =====
-- add_column(:memos, :category_id, :integer)
   -> 0.0010s
== AddCategoryIdToMemos: migrated (0.0444s) =====

[root@cisnote memopad]#
```

Confirm the result of migration

After you have migrated the Column Adding change, confirm the result by typing sqlite3 command.

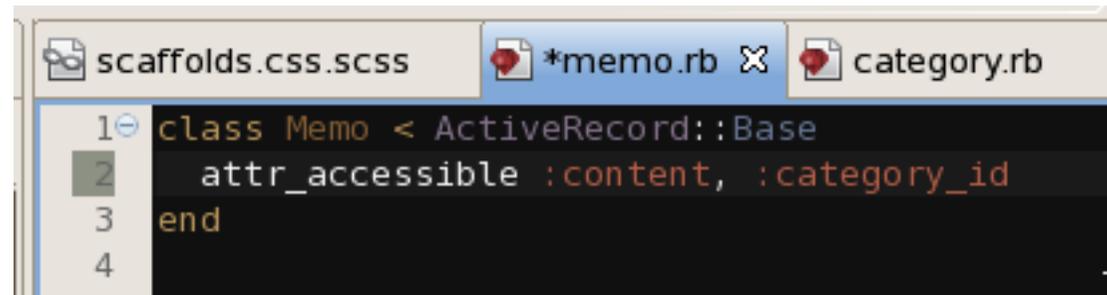
```
[root@cisnote memopad]# cd db
[root@cisnote db]# ls
development.sqlite3  migrate  schema.rb  seeds.rb
[root@cisnote db]# sqlite3 development.sqlite3
SQLite version 3.3.6
Enter ".help" for instructions
sqlite> .schema memos
CREATE TABLE "memos" ("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "content" text,
"created_at" datetime NOT NULL, "updated_at" datetime NOT NULL, "category_id" integer
);
sqlite> .exit
[root@cisnote db]# █
```

Set category_id accessible

- ❑ In memopad/app/models/memo.rb

```
class Memo < ActiveRecord::Base
  attr_accessible :content, :category_id
end
```

- ❑ Add :category_id to the list of accessible attributes.



```
scaffolds.css.scss *memo.rb category.rb
1 class Memo < ActiveRecord::Base
2   attr_accessible :content, :category_id
3 end
4
```

One to Many Relationship

- From memos to categories, choose one.
- From categories, there are many memos in one category.

In memo model (app/models/memo.rb)

`belongs_to :category`

In category model (app/models/category.rb)

`has_many :memos`

Be careful of singular/plural form.

Setting up relationship

□ memopad/app/models/categories.rb

```
class Category < ActiveRecord::Base
```

```
  has_many :memos
```

```
end
```

```
1 class Category < ActiveRecord::Base
2   attr_accessible :name
3   has_many :memos
4 end
5
```

□ memopad/app/models/memo.rb

```
class Memo < ActiveRecord::Base
```

```
  belongs_to :category
```

```
end
```

```
1 class Memo < ActiveRecord::Base
2   attr_accessible :content, :category_id
3   belongs_to :category
4 end
5
```

When you mistyped

- ❑ When you mistyped in scaffolding, then type rails destroy (something you have specified.)
ex. rails destroy scaffold Category
rails destroy migration AddCategoryIdToMemos
- ❑ If you successfully generated scaffold, and run the following already;
rake db:migrate
- ❑ If you noticed your mistakes after the migration, type rollback after the successful migration.
rake db:rollback

One to One, Many to Many cases

- In case of One to One, either one should become main.
 - Main model `has_one`
 - Sub model `belongs_to`

- In case of Many to Many relationship,
 - both `has_and_belongs_to_many`

Singular form or plural form?

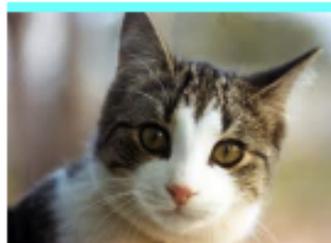
- `belongs_to, has_one(space):(singular)`
- `has_many, has_and_belongs_to_many cases, (space):(colon)(plural form)`
- Ruby has method to convert singular form into plural form, and plural form into singular form.
 - Please see Page 47 of day 6 slides.

Type of relations between tables

- One to one
 - Student number $\langle == \rangle$ Student Name
- One to many
 - Name in address book $\langle == \rangle$ Phone numbers
- Many to many
 - Guest in restaurant $\langle == \rangle$ ordered dishes

Today's goal

We are going to get the following screen;
Here we have a drop down list of categories
in the memo's input screen.



[Listing memos](#) | [New Memo](#) | [Ruby Official Site](#) | [My Theme](#)

Listing memos

Content	Category	
Yesterday, it was rainy because of the typhoon.	Idea	Show E

[New Memo](#)

Category

[Show](#) | [Back](#)

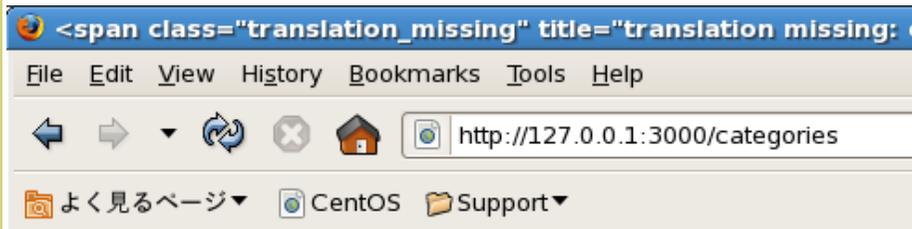
Preparation for the last modification

When we miss the link, it will cause an error in tracing the link from memos to categories.

So, make sure that we have 'category_id' field values in all memos.

Make sure we have categories list

Like memos listing screen, we have whole set of screens to add, edit, show and remove the categories.



[Listing memos](#) | [New Memo](#) | [Ruby Official Site](#) | [My Theme Page\(Prepa](#)

Listing categories

Name

[New Category](#)



[Listing memos](#) | [New Memo](#) | [Ruby Official Site](#) | [M](#)

Listing categories

Name

To Do [Show](#) [Edit](#) [Destroy](#)

Idea [Show](#) [Edit](#) [Destroy](#)

Phone message [Show](#) [Edit](#) [Destroy](#)

Dinner [Show](#) [Edit](#) [Destroy](#)

[New Category](#)

Add Category_id input in _form

Edit `app/views/memos/_form.html.erb`

Add the following four lines;

```
<div class="field">
```

```
  <%= f.label :category_id %><br />
```

```
  <%= f.text_field :category_id %>
```

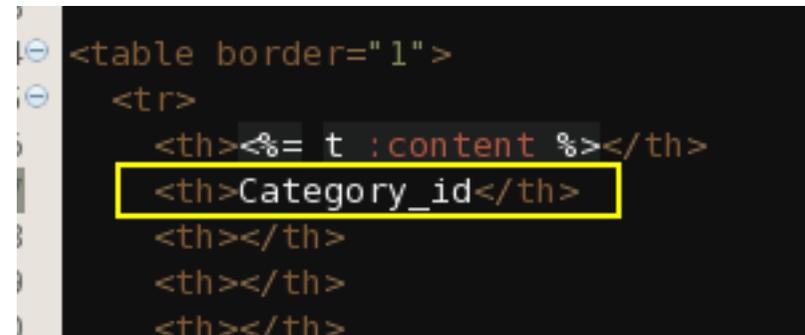
```
</div>
```

```
13
14⊖ <div class="field">
15   <%= f.label :content %><br />
16   <%= f.text_area :content %>
17 </div>
18⊖ <div class="field">
19   <%= f.label :category_id %><br />
20   <%= f.text_field :category_id %>
21 </div>
22⊖ <div class="actions">
23   <%= f.submit %>
```

Add category_id in index.html.erb

Below the `t :content` line, add `Category_id` heading.

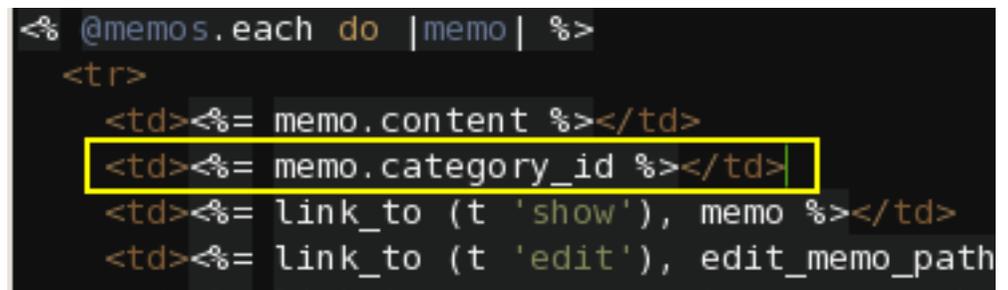
```
<tr>
  <th><%= t :content %></th>
  <th>Category_id</th>
  <th></th>
  <th></th>
```



```
<table border="1">
  <tr>
    <th><%= t :content %></th>
    <th>Category_id</th>
    <th></th>
    <th></th>
```

Then, add `<%= memo.category_id %>` below content.

```
<td><%= memo.content %></td>
<td><%= memo.category_id %></td>
<td><%= link_to (t 'show'), memo %></td>
```



```
<%= @memos.each do |memo| %>
  <tr>
    <td><%= memo.content %></td>
    <td><%= memo.category_id %></td>
    <td><%= link_to (t 'show'), memo %></td>
    <td><%= link_to (t 'edit'), edit_memo_path
```

Make sure all memos have category_id

In order to avoid link error, just make sure that all memos have category_id listed on the screen.



[Listing memos](#) | [New Memo](#) | [Ruby Official Site](#) | [My Theme Page\(Preparing\)](#)

Listing memos

Content	Category_id			
Yesterday, it was rainy because of the typhoon.	1	Show	Edit	Destroy

[New Memo](#)

Drop Down list

Modify app/views/memos/_form.html.erb

Now coment out f.text_field :category_id,
and then add the following line.

```
<%= f.select :category_id, Category.all  
  .collect{|c|[c.name,c.id]} %>
```

```
13  
14⊖ <div class="field">  
15   <%= f.label :content %><br />  
16   <%= f.text_area :content %>  
17 </div>  
18⊖ <div class="field">  
19   <%= f.label :category_id %><br />  
20   <%= f.select :category_id, Category.all.collect{|c|[c.name,c.id]} %>  
21   <!-- <%= f.text_field :category_id % -->  
22 </div>  
23⊖ <div class="actions">  
24   <%= f.submit %>  
25 </div>
```

Category

Idea

Update Memo

Show | Back

Modify List from ID to Link

Modify `app/views/memos/index.html.erb`

Replace

```
<th>Category_id</th>
```

```
<td><%= memo.category_id %></td>
```

To

```
<th>Category</th>
```

```
<td><%= memo.category.name %></td>
```

```
12
13 <%= @memos.each do |memo| %>
14 <tr>
15   <td><%= memo.content %></td>
16   <td><%= memo.category.name %></td>
17   <td><%= link_to (t 'show'), memo %></td>
18   <td><%= link_to (t 'edit'), edit_memo_path
19   <td><%= link_to (t 'destroy'), memo, metho
```

Listing Memos Screen

Now the screen should be like this;



Listing memos

Content	Category	
Yesterday, it was rainy because of the typhoon.	Idea	Show E

[New Memo](#)

Link of Relation

Once we embed the link by writing `relation_id` in the field list, e.g. `category_id`, then, we can access to the any linked field by writing `relation.field_name`, e.g. `category.name`.

See Console Screen

We can check the Database Access, by reading SQL display in the console.

```
Started GET "/memos" for 127.0.0.1 at 2012-11-07 10:18:07 +0900
Processing by MemosController#index as HTML
  Memo Load (0.2ms)  SELECT "memos".* FROM "memos"
  Category Load (0.1ms)  SELECT "categories".* FROM "categories" WHERE "categories"."id" = 2 LIMIT 1
  Rendered memos/index.html.erb within layouts/application (2.2ms)
  Rendered shared/_menu_bar.html.erb (0.7ms)
  Rendered shared/_right_bar.html.erb (0.0ms)
Completed 200 OK in 11ms (Views: 9.2ms | ActiveRecord: 0.4ms)
```

Report Theme

- No report is required, for today.

Absence Report

List up all the changes you have done to let memos have the link to categories table.

Then, add the 'Listing Memos' Screen you have updated so that category of each memo shows the category name.