

# Model CRUD

# Insert (I)

- save
- save!
- create
- create!

# Insert (II)

- 利用Rails Console來測試：

```
group = Group.new(:name => "test group")
group.save
group.save!
group = Group.new(:name => "test")
group.save
group.save!
group.save(:validate => false)
```



# 新增 (III)

- 利用Rails Console來測試：

```
group = Group.create(:name => "test group1")  
group = Group.create!(:name => "test group2")  
group = Group.create(:name => "test")  
group = Group.create!(:name => "test")
```

# 一般查詢 (I)

- find(id)
- find([id1, id2])
- first
- last
- all

# 一般查詢 (II)

- `find_by_{column_name}`
- `find_by_sql`



# 一般查詢 (III)

- 利用Rails Console來測試：

```
Group.find(1)
```

```
Group.find(1000)
```

```
Group.all
```

```
Group.find_by_sql("select * from groups")
```

```
Group.find_by_name("test group")
```

```
Group.where(name: "test group").first
```

# 組合查詢(I)

- where
  - `Group.where(:name => "test group")`
  - `Group.where(["name = ?", "test group"])`
  - `Group.where("name = '#{name}')` #不要這樣寫
- limit
  - `Group.limit(3)`



# 組合查詢(II)

- order
  - `Group.limit(5).order("id DESC")`
- offset
  - `Group.limit(5).order("id DESC").offset(2)`
- select
  - `Group.limit(5).select("id name")`

# 組合查詢(III)

- readonly
  - `group = Group.readonly.first`
  - `group.update_attribute(:name, "test group100")` # Error
- Group
  - `Group.order("date(updated_at)").limit(5)`
- Having

# 更新

- `update_attribute`
  - `Group.first.update_attribute(:name, "group1")`
  - `Group.first.update_attribute(:name, "grp")`
- `update_attributes`
  - `Group.first.update(:name => "group2")`
  - `Group.first.update(:name => "grp")`



# 更新

- `update_attribute`
  - `Group.first.update_attribute(:name, "group1")`
  - `Group.first.update_attribute(:name, "grp")`
- `update_attributes`
  - `Group.first.update(:name => "group2")`
  - `Group.first.update(:name => "grp")`

注意：

`update_attribute` 會略過 `validation` 資料驗證

# 删除

- Model#destroy
  - Group.first.destroy
- Model.delete(id)
  - Group.delete(2)
- Model.delete\_all
  - Group.delete\_all

# 統計

- Model.count
  - Group.count
- Model.average(:column)
- Model.maximum(:column)
  - Group.maximum(:name)
- Model.sum(:column)



# Scope (I)

- 宣告常用的查詢條件，讓程式乾淨易讀。

```
class Group < ActiveRecord::Base
  #...
  scope :recent, ->{where("updated_at > ?",
                          Time.now - 1.day).limit(5)}
  #...
end
```

# Scope (II)

- `default_scope`: 預設的搜尋條件

```
class Group < ActiveRecord::Base
  #...
  default_scope order("id DESC")
  #...
end
```

# Scope(III)

- 也可以接受參數

```
class Group < ActiveRecord::Base
  #..
  scope :recent, ->(data){
    where("updated_at > ?", date).limit(5)
  }
  #...
end
```

Group.recent(Time.now - 1.day)

建議使用class method來實作