1. 在Jupyter notebook中操作學習ORM

需學習ORM，具備功能較多

參看課堂範例

1. ChatGPT的ORM教學

迅速鳥瞰一下!

<https://chatgpt.com/share/680e31cf-c0a0-8001-a4c5-64c7d97f6248>

1. Django **CRUD（增刪改查）**

下面是 Django **CRUD（增刪改查）** 的完整範例，  
每一個操作都會給你兩種做法：  
（1）用 **Django ORM**，（2）用 **純 SQL + cursor**。

假設有一個 Model 叫做：

python

複製編輯

# models.py

from django.db import models

class Book(models.Model):

title = models.CharField(max\_length=100)

author = models.CharField(max\_length=100)

## 1. ****Create 新增****

### ORM 做法

python

複製編輯

book = Book.objects.create(title="Django Basics", author="Tom")

### 純 SQL 做法

python

複製編輯

from django.db import connection

with connection.cursor() as cursor:

cursor.execute(

"INSERT INTO myapp\_book (title, author) VALUES (%s, %s)",

["Django Basics", "Tom"]

)

## 2. ****Read 讀取****

### ORM 做法

python

複製編輯

books = Book.objects.all()

for book in books:

print(book.title, book.author)

### 純 SQL 做法

python

複製編輯

from django.db import connection

with connection.cursor() as cursor:

cursor.execute("SELECT id, title, author FROM myapp\_book")

rows = cursor.fetchall()

for row in rows:

print(row)

## 3. ****Update 更新****

### ORM 做法

python

複製編輯

book = Book.objects.get(id=1)

book.title = "Django Advanced"

book.save()

### 純 SQL 做法

python

複製編輯

from django.db import connection

with connection.cursor() as cursor:

cursor.execute(

"UPDATE myapp\_book SET title = %s WHERE id = %s",

["Django Advanced", 1]

)

## 4. ****Delete 刪除****

### ORM 做法

python

複製編輯

book = Book.objects.get(id=1)

book.delete()

### 純 SQL 做法

python

複製編輯

from django.db import connection

with connection.cursor() as cursor:

cursor.execute("DELETE FROM myapp\_book WHERE id = %s", [1])

### 小提醒：

* ORM 方法，**有自動防止 SQL Injection**，而且寫法更簡單。
* 純 SQL 方法要自己小心參數綁定（記得用 %s + 參數，不要直接字串拼接）。
* myapp\_book 是 app\_name + \_ + model\_name，都是小寫！  
  （這是 Django 自動產生的 table name）

1. 常用指令

這邊我幫你整理出一個「**Django ORM vs 純 SQL 對照表（常用版）**」，超清楚：

| **功能** | **ORM 做法** | **純 SQL 做法** |
| --- | --- | --- |
| **新增一筆** | Book.objects.create(title="A", author="Tom") | INSERT INTO myapp\_book (title, author) VALUES (%s, %s) |
| **查全部** | Book.objects.all() | SELECT \* FROM myapp\_book |
| **查一筆 (用 ID)** | Book.objects.get(id=1) | SELECT \* FROM myapp\_book WHERE id = %s |
| **條件查詢** | Book.objects.filter(author="Tom") | SELECT \* FROM myapp\_book WHERE author = %s |
| **更新一筆 (先查再存)** | book = Book.objects.get(id=1) book.title = "New" book.save() | UPDATE myapp\_book SET title = %s WHERE id = %s |
| **批次更新** | Book.objects.filter(author="Tom").update(title="New Title") | UPDATE myapp\_book SET title = %s WHERE author = %s |
| **刪除一筆 (先查再刪)** | book = Book.objects.get(id=1) book.delete() | DELETE FROM myapp\_book WHERE id = %s |
| **批次刪除** | Book.objects.filter(author="Tom").delete() | DELETE FROM myapp\_book WHERE author = %s |
| **只選某些欄位** | Book.objects.values("title", "author") | SELECT title, author FROM myapp\_book |
| **排序** | Book.objects.order\_by('title') | SELECT \* FROM myapp\_book ORDER BY title ASC |
| **限制筆數** | Book.objects.all()[:10] | SELECT \* FROM myapp\_book LIMIT 10 |

### 小補充

* ORM .filter() 是拿到**多筆**，要搭配 .first() 或 .last() 才只取一筆。
* ORM .get() 直接拿一筆，但如果找不到 or 太多筆，會丟錯誤。
* SQL 自己控制，記得 WHERE 和 LIMIT。