

- Ex3-4Add1:請問什麼是「物件關聯對映」(ORM, Object Relational Mapping), 主要的優點與缺點是?
 - Ex3-4Add2:ORM 可幫使用者用 python 寫法來執行 sql, 是否也有不需要 ORM 來建立資料庫的方法?
 - Ex3-4Add3:資料庫有四種常用函式: 新增(Create), 讀取(Read), 更新(Update), 刪除>Delete); 通常會被簡寫為 CRUD, 用請由這四個方向, 配合 SQL 的指令, 來簡單說明 SQLAlchemy 會是如何?
 - Ex3-4Add4:透過 SQLAlchemy 操作資料庫時, 所有數據庫插入和刪除操作均在 Python 中進行。只有當我們使用什麼方法完成之後, 數據庫才會被更改?這個操作一般是儘量越少越好, 理由是?
 - Ex3-4Add5:為了方便查閱, 請設計在透過 SQLAlchemy 操作資料庫時可以將「目前所查到的資料物件」轉為 dict 的副程式。並簡單示範使用方法。
 - Ex3-4Add6:知名的網站框架 Django, 也有 ORM, (a)若 ORM 的基類(base class)在 Django 是 models.Model, 請問在 Django 的可能程式寫法是什麼? (a)另外在開發 Django 之前, 若已有現存舊資料庫, 是否有工具幫忙轉成符合 ORM 精神的程式碼?
 - Ex3-4Add7:為了像 Python 的寫法來寫 sql, 已知 SQLAlchemy 要繼承的基類是 Base, 請以 SQLAlchemy 為例, 設計一個可以產生表格名稱稱為 winners, 欄位至少有一個 id 的 ORM 程式。只要前幾行能表達正確想法即可。
 -
-

Ex3-4Add1:請問什麼是「物件關聯對映」(ORM, Object Relational Mapping), 主要的優點與缺點是?

Ans:

ORM 在網站開發結構, 是位於『資料庫』和『Model 資料容器』兩者之間的一項技術。是直接利用 Python 的語法對資料庫進行操作, 不需要直接寫 SQL 語法, ORM 背後會自動將 Python 代碼轉換成應對的 SQL 語法, 再來進行對資料庫的操作。

目的是幫助使用者簡便、安全完成資料庫讀取。

- 優點: 可讀性較高, 也可以防止 SQL injection
- 缺點: 因為多了一層 python 的包裝會犧牲掉部分的效能

Ex3-4Add2:ORM 可幫使用者用 python 寫法來執行 sql, 是否也有不需要 ORM 來建立資料庫的方法?

Ans:

至少還有兩種: (1)直接對連接好的資料庫寫出 Sql 指令 (2)或是在圖形界面建立表格結構或增減資料。

Ex3-4Add3:資料庫有四種常用函式: 新增(Create), 讀取(Read), 更新(Update), 刪除>Delete); 通常會被簡寫為 CRUD, 用請由這四個方向, 配合 SQL 的指令, 來簡單說明 SQLAlchemy 會是如何?

Ans:

新增(Create): 在 SQLAlchemy 是直接準備好字典物件, 用 session.add(Winner)的方法新增。一般的 SQL 是用 "INSERT INTO "當作命令。

讀取(Read): 用 session.query(條件)查詢後, 即可讀取資料而查詢的 SQL: 例如: WHERE

winners.nationality = ?

更新(Update): 更新的 SQL 是在查詢後，可更新(update)查詢結果，

例如：UPDATE winners SET nationality=? WHERE winners.id =?

刪除(Delete): 刪除的 SQL 是在查詢後，可以刪除(delete)查詢結果，例如：DELETE FROM winners WHERE winners.name = ?

Ex3-4Add4:透過 SQLAlchemy 操作資料庫時，所有數據庫插入和刪除操作均在 Python 中進行。只有當我們使用什麼方法完成之後，數據庫才會被更改?這個操作一般是儘量越少越好，理由是？

Ans:

(a)提交(Commit)

(b)關於 commit, 因為提交(commit)，會涉及建立數據庫握手和協商交易(negotiating transaction)，通常是一個緩慢的過程，應該盡可能減少提交(commit)次數，使 SQLAlchemy 在幕後發揮作用。

Ex3-4Add5:為了方便查閱，請設計在透過 SQLAlchemy 操作資料庫時可以將「目前所查到的資料物件」「轉為 dict」的副程式。並簡單示範使用方法。

Ans:

在 SQLAlchemy__將目前查到的數據物件轉為 dict

#Example 3-4. Converts an SQLAlchemy instance to a dict

def inst_to_dict(inst, delete_id=True):

 dat = {}

 for column in inst.__table__.columns:

 dat[column.name] = getattr(inst, column.name)

 if delete_id:

 dat.pop('id') #If delete_id is true, remove the SQL primary ID field.

 return dat

winner_rows = session.query(Winner)

nobel_winners_2 = [inst_to_dict(w) for w in winner_rows]

print(nobel_winners_2)

Ex3-4Add6:知名的網站框架 Django，也有 ORM，(a)若 ORM 的基類(base class)在 Django 是 models.Model，請問在 Django 的可能程式寫法是什麼？(a)另外在開發 Django 之前，若已有現存舊資料庫，是否有工具幫忙轉成符合 ORM 精神的程式碼？

Ans:

(a)ORM 的基類(base class)_在 Django 的基類是 models.Model

所以 ORM 的可能寫法如下：

from django.db import models

class table_my(models.Model):

 text=models.CharField(max_Length=200)

...

(b)Django 內建可以「反向」找出底層資料庫模型的 ORM 程式碼的工具名稱是

inspectdb 管理命令，這個工具目的是對現有舊資料庫，生成 ORM 模型程式代碼的工具。

Ex3-4Add7:為了像 Python 的寫法來寫 sql,已知 SQLAlchemy 要繼承的基類是 Base, 請以 SQLAlchemy 為例, 設計一個可以產生表格名稱為 winners, 欄位至少有一個 id 的 ORM 程式. 只要前幾行能表達正確想法即可。

Ans:

```
from sqlalchemy import Column, Integer, String, Enum
class Winner(Base): ##### 用 python 的物件寫法, 寫出 sql 建立的表格綱要(table schema)的敘述
    __tablename__ = 'winners'    ##●●● table 的名稱
    id = Column(Integer, primary_key=True)    ###第 0 項 (key)項資料行(為了當作索引 key 的序號!)
    name = Column(String)                ###第 1 項資料行 name
    category = Column(String)
    year = Column(Integer)
    nationality = Column(String)
    sex = Column(Enum('male', 'female'))
```
