

目錄

Install Document	2
01 複製既有的遠端數據庫.....	2
02 查看遠端數據庫分支的修改內容	2
03 合併分支.....	2
04 更新本機上檔案	2
05 位址轉到專案	3
06 修改環境參數.....	3
07 Model Migration	3
08 執行伺服器	5
09 申請 MySQL / ngrok	5
10 執行 ngrok.....	5
11 新增使用者 Creating users.....	6
Project Document.....	7
01 GitHub	7
02 顯示所有的 Table 列表	7
03 資料集.....	7
04 使用 Table	10
05 查看特定 Table.....	10
06 新增 Table	10
07 推薦清單資料庫欄位	10
08 更改 CODE.....	10
MySQL Document.....	12
01 SQL 語法	12
02 Django 關聯 MySql 數據庫.....	13
03 實戰演練.....	14
04 問題解決.....	17

Install Document

01 複製既有的遠端數據庫

`λ git clone <url>`

02 查看遠端數據庫分支的修改內容

`λ git fetch <repository>`

03 合併分支

`λ git stash`

```
D:\中興資管所\7 實驗進度\musicPlatform\m.platform (master)
λ git stash
warning: LF will be replaced by CRLF in other/SQLTables/tracks_album.sql.
The file will have its original line endings in your working directory
Saved working directory and index state WIP on master: b428294 Merge branch 'master' of ht
tps://github.com/plusoneee/m.platform

D:\中興資管所\7 實驗進度\musicPlatform\m.platform (master)
λ git pull
Updating b428294..b73c507
Fast-forward
 tracks/urls.py | 7 +++++
 tracks/views.py | 36 ++++++
 web/templates/madeForYou.html | 43 ++++++
 web/templates/madeForYouChoose.html | 30 ++++++
 web/templates/navbar.html | 6 +++
 5 files changed, 116 insertions(+), 6 deletions(-)
 create mode 100644 web/templates/madeForYou.html
 create mode 100644 web/templates/madeForYouChoose.html
```

04 更新本機上檔案

`λ git pull <repository 網址>`

遇到錯誤：fatal: not a git repository (or any of the parent directories): .git

解決方法：λ git init

更新的檔案

```
D:\中興資管所\7 實驗進度\musicPlatform\m.platform (master)
λ git pull https://github.com/plusoneee/m.platform.git
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 4), reused 6 (delta 4), pack-reused 0
Unpacking objects: 100% (6/6), done.
From https://github.com/plusoneee/m.platform
 * branch          HEAD      -> FETCH_HEAD
Updating 549665a..b428294
Fast-forward
 tracks/models.py | 20 ++++++
 1 file changed, 18 insertions(+), 2 deletions(-)
```

本機 > DATA (D:) > 中興資管所 > 7 實驗進度 > musicPlatform > m.platform > tracks

名稱	修改日期	類型	大小
__pycache__	2019/6/17 下午 03:32	檔案資料夾	
migrations	2019/6/17 下午 03:38	檔案資料夾	
Module	2019/6/17 下午 03:32	檔案資料夾	
__init__.py	2019/6/17 下午 03:15	PY 檔案	0 KB
admin.py	2019/6/17 下午 03:15	PY 檔案	1 KB
apps.py	2019/6/17 下午 03:15	PY 檔案	1 KB
models.py	2019/6/17 下午 08:15	PY 檔案	3 KB
tests.py	2019/6/17 下午 03:15	PY 檔案	1 KB
urls.py	2019/6/17 下午 03:15	PY 檔案	2 KB
views.py	2019/6/17 下午 03:15	PY 檔案	7 KB

05 位址轉到專案

移到 D 槽 → `λ d:`

輸入位址 → `λ cd D:\中興資管所\7 實驗進度\musicPlatform\m.platform`

06 修改環境參數

執行 `cp web/my.example.cnf web/my.cnf`

或者於 web 資料夾複製 my.example.cnf 修改檔案之資料庫連接帳號密碼，另存為 my.cnf

```
my.example.cnf - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
[client]
database = tracks
user = 'ur_mysql_user_name'
password = 'ur_mysql_password'
default-character-set = utf8
```

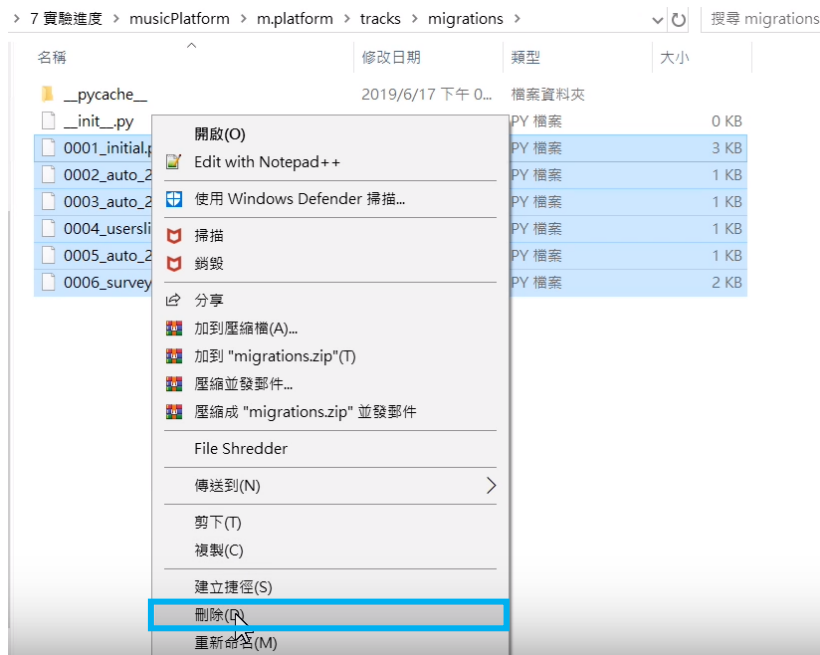
07 Model Migration

`λ python manage.py makemigrations tracks` (tracks 可有可無)

移至 `D:\中興資管所\7 實驗進度\musicPlatform\m.platform\tracks\migrations`

刪除底下 0001_initial.py ... 等檔案

留下「__pycache__ 資料夾」和「__init__.py 檔案」



正確結果：

```
D:\中興資管所\7 實驗進度\musicPlatform\m.platform (master)
λ python manage.py makemigrations tracks
Migrations for 'tracks':
  tracks\migrations\0001_initial.py
    - Create model Album
    - Create model Artist
    - Create model Features
    - Create model RecSysResults
    - Create model Survey20Tracks
    - Create model SurveyResults
    - Create model UsersLike
    - Create model UserSurveyCompleted
```

λ python manage.py migrate

```

[?] 選取 命令提示字元 - python manage.py migrate
D:\中興資管所\7 實驗進度\musicPlatform\m.platform>python manage.py makemigrations
Migrations for 'tracks':
  tracks\migrations\0001_initial.py
    - Create model Album
    - Create model Artist
    - Create model Features
    - Create model Survey20Tracks
    - Create model SurveyResults
    - Create model UsersLike
    - Create model UserSurveyCompleted
    - Add field artist to album

D:\中興資管所\7 實驗進度\musicPlatform\m.platform>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, tracks
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK

```

08 執行伺服器

λ python manage.py runserver

網址要補上/tracks → http://127.0.0.1:8000/tracks

```
D:\中興資管所\7 實驗進度\musicPlatform\m.platform (master)
λ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
June 18, 2019 - 14:01:19
Django version 2.1, using settings 'web.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

09 申請 MySQL / ngrok

MySQL

帳號：root / chiouchingyi (擇一)

密碼：850121

ngrok

帳號：chiouchingyi0121@gmail.com

密碼：jenny850121

10 執行 ngrok

A. 點擊 ngrok.exe 執行檔

D:\程式檔\ngrok.exe

B. 連接至帳號 Connect your account (到 ngrok 網站 Auth 查看)

Your Tunnel Authtoken : 7v3SqyXN25gZEtqSKrV9H_4hphM6C4bNgyWCFct5RG4

C. 執行檔執行指令

λ ngrok authtoken 7v3SqyXN25gZEtqSKrV9H_4hphM6C4bNgyWCFct5RG4

D. 執行 HTTP tunnel 在自己的 port

λ ngrok.exe http 8000 (藍框處)

① 127.0.0.1:8000/tracks/browser/artist/

E. 修改 setting.py 檔案

D:\中興資管所\7 實驗進度\musicPlatform\m.platform\web\setting.py

ALLOWED_HOSTS = [] 改為 ALLOWED_HOSTS = ['*']

<https://docs.djangoproject.com/en/2.2/ref/settings/#allowed-hosts>

F. 複製新網址 Forwarding

要隨時更新網址 (黃底處會更新)

<http://d8ec108a.ngrok.io/tracks> (要加上/tracks)

```
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             chioungyi (Plan: Free)
Version             2.3.29
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://d8ec108a.ngrok.io -> http://localhost:8000
Forwarding           https://d8ec108a.ngrok.io -> http://localhost:8000

Connections          ttl    opn    rt1    rt5    p50    p90
                   19     0     0.00   0.00   0.34   0.96

HTTP Requests
-----
GET /static/css/forBrowserArtist.css 200 OK
GET /tracks/browser/artist/          200 OK
GET /tracks/                          302 Found
POST /accounts/login/                302 Found
GET /favicon.ico                     404 Not Found
GET /accounts/login/                 200 OK
GET /tracks/                          302 Found
GET /favicon.ico                     400 Bad Request
GET /tracks/                          400 Bad Request
GET /favicon.ico                     400 Bad Request
```

11 新增使用者 Creating users

Django Document : <https://docs.djangoproject.com/en/2.2/topics/auth/default/>

A. Cmder 執行 manage.py 檔

λ python manage.py shell

```
D:\中興資管所\7 實驗進度\musicPlatform\m.platform (master)
λ python manage.py shell
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> |
```

B. 輸入 create 指令

修改粗體紅字處，設定帳密相同(方便起見)

```
>>> from django.contrib.auth.models import User
>>> user = User.objects.create_user('john', 'email@example.com', 'johnpassword')
>>> user.save()
```

```
>>> from django.contrib.auth.models import User
>>> user = User.objects.create_user('test', 'lennon@thebeatles.com', 'test')
>>> user.save()
```

C. 重新啟動伺服器

λ python manage.py runserver

```
D:\中興資管所\7 實驗進度\musicPlatform\m.platform (master)
λ python manage.py runserver
```

網址要補上/tracks → <http://127.0.0.1:8000/tracks>

Project Document

01 GitHub

<https://github.com/plusoneee/m.platform>

02 顯示所有的 Table 列表

`show tables from tracks;` (tracks 底下的 Table)

Tables_in_tracks	Table 說明 (包含的欄位查看檔案 model.py)
auth_group	
auth_group_permissions	
auth_permission	
auth_user	建立的使用者
auth_user_groups	
auth_user_user_permissions	
django_admin_log	
django_content_type	
django_migrations	
django_session	
tracks_album	專輯
tracks_artist	歌手
tracks_audiolyricembedding	16517 首音頻歌詞特徵
tracks_audiolyricstop20	16517 首對應前 20 相似歌曲
tracks_features	歌曲特徵
tracks_recfromrlresults	推薦結果(RL)
tracks_recsysresults	推薦結果(content based)
tracks_survey20tracks	調查的 20 首歌曲
tracks_survey20tracksfeatures	調查的 20 首歌曲特徵
tracks_surveyresults	調查結果
tracks_userslike	使用者喜歡清單(加愛心)
tracks_usersurveycompleted	使用者是否填過調查問卷

03 資料集

A. 歌手資料集

tracks_artist.csv

Table 名稱: tracks_artist

id	name	genres	img_url
歌手 ID	歌手名稱	類型	歌手圖片

<https://open.spotify.com/artist/5Z1CCuBsyhEHngq3U5IraY>

B. 專輯資料集

tracks_album.csv

Table 名稱：tracks_album

id	name	preview_url	img_url	tracks_number	release_date	artist_id
專輯 ID	專輯名稱	專輯網址	專輯圖片	專輯歌曲數量	發行日期	歌手 ID

<https://open.spotify.com/album/22pfCAdDOgyC2JfSi5OwxT>

C. 歌曲特徵資料集

tracks_features.csv

Table 名稱：tracks_features

id	name	preview_url	img_url	tracks_number	release_date	artist_id
專輯 ID	專輯名稱	專輯網址	專輯圖片	專輯歌曲數量	發行日期	歌手 ID

D. 音頻歌詞向量資料集

Table 名稱：tracks_audiolyricembedding

id	name	audio_vector	lyrics_vector
歌曲 ID	歌曲名稱	音頻向量	音頻向量

E. 音頻歌詞前 20 首相似資料集

Table 名稱：tracks_audiolyricstop20

id	a_top1	...	a_top20	l_top1	...	l_top20
歌曲 ID	歌曲 ID	...	歌曲 ID	歌曲 ID	...	歌曲 ID

F. 推薦歌曲資料集

trakcs_data_for_rec.csv

name	id	artist_name	track_preview_url	artist_img_url	album_name	album_id	img_url
歌曲名稱	歌手 ID	歌手名稱	歌曲網址	歌手圖片	專輯名稱	專輯 ID	專輯圖片

G. 調查歌曲資料集

tracks_survey20tracks.csv

Table 名稱：tracks_survey20tracks

name	id	artist_name	artist_id	preview_url
歌曲名稱	歌曲 ID	歌手名稱	歌手 ID	歌曲網址

H. 調查歌曲特徵資料集

tracks_survey20tracks_vector(edit).csv

Table 名稱：tracks_survey20tracksfeatures

id	name	audio_vector	lyrics_vector
歌曲 ID	歌曲名稱	音頻向量	歌詞向量

I. 調查結果

Table 名稱：surveyresults

id	user	results	date
編號	用戶 ID	調查分數	調查時間

J. 推薦結果(content based)

Table 名稱：tracks_recsysresults_a(音頻)/l(歌詞)/h(混合)

id	album_img_url	track_id	artist_name	recom_method	recom_rank	socre
用戶 ID	專輯圖片	歌曲 ID	歌手名稱	推薦方法	推薦排序	分數

recom_method : 0: Audio, 1: Lyrics, 2: Hybrid

K. 推薦結果(RL)

Table 名稱：tracks_recfromrlresults

id	album_img_url	track_id	artist_name	recom_rank	socre	number_of_rec_times
用戶 ID	專輯圖片	歌曲 ID	歌手名稱	推薦排序	分數	推薦次數

L. 用戶喜歡歌曲

Table 名稱：tracks_userslike

id	user	track_id	album_id
編號	用戶 ID	歌曲 ID	專輯 ID

M. 調查完成

Table 名稱：tracks_usersurveycompleted

id	user	complete_date
編號	用戶 ID	完成時間

N. 資料缺失

- track_album.csv

專輯 ID 52O1oc2EC6kwNo8iwBs919 → 沒有專輯圖片

專輯 ID 70HXQp5RAjM37JHBuvL8Dg → 沒有專輯圖片

- trakcs_data_for_rec.csv

專輯 ID 70HXQp5RAjM37JHBuvL8Dg → 沒有專輯圖片

- tracks_data_for_rec.csv

專輯 ID 70HXQp5RAjM37JHBuvL8Dg → 沒有專輯圖片

04 使用 Table

use tracks;

05 查看特定 Table

```
select * from tracks_artist;
select * from tracks_features;
select * from tracks_survey20tracks;
select * from auth_user;
```

06 新增 Table

修改 model.py 檔案

新增一個 class _____(models.Model):

```
class SongFeatures(models.Model):
    id = models.CharField(max_length=30, primary_key=True) # track id
    name = models.CharField(max_length=100)
    audio_features = models.FloatField ()
    lyrics_features = models.FloatField ()
```

07 推薦清單資料庫欄位

- 資料表：tracks_recsysresults
- 推薦候選歌曲 16517 首 csv_20190615/trakcs_data_for_rec.csv
- 欄位包括：
 - user_id、album_img_url、track_id、artist_id、recom_method、recom_rank、score
 - recom_method：推薦的方法 (0: Audio, 1: Lyrics, 2: Hybrid, 3: RL)
 - recom_rank：推薦順序
 - score：使用者給推薦歌曲的分數(1-5 分)
- 欄位資料從其他 Table 查詢
 - track_id → (對照 Table tracks_features) → album_id、artist_id
 - album_id → (對照 Table tracks_album) → album_img_url

08 推薦方法

A. 實驗

- 修改使用者
 - user_id = 'chiouchingyi'
- 修改推薦方法
 - recom_method = "0" #Audio
 - recom_method = "1" #Lyrics
 - recom_method = "2" #Hybrid
- 修改路徑
 - 音頻推薦(a)、歌詞推薦(l)、混合推薦(h)

```
filepath = data_folder + user_id + '_surveyresult_a' + '.csv'
filepath2 = data_folder + user_id + '_surveyresult_a_new' + '.csv'
```

09 更改 CODE

原始	更改
ALLOWED_HOSTS = []	ALLOWED_HOSTS = ['*']
USE_TZ = True	##USE_TZ = True (註解掉)

MySQL Document

01 SQL 語法

建立資料庫	create database 資料庫名稱;	
列出所有資料庫	show databases;	
刪除資料庫	drop database 資料庫名稱;	
使用資料庫	use 資料庫名稱;	
建立資料表	create table 資料表名稱(id integer auto_increment primary key, name char(20), mail char(50), home char(50), messages char(50));	常用資料庫資料型態 A. INT (整數) B. CHAR (1~255 字元字串) C. VARCHAR (不超過 255 字元不定長度字 串) D. TEXT (不定長度字串最多 65535 字元)
列出資料表欄位資訊	describe 資料表名稱;	
刪除資料表	drop table 資料表名稱;	
清空資料表	truncate table 資料表名稱; (只清除資料並保留結構、欄位、索引)	
查詢單一欄位資料	select 欄位名 from 資料表名稱;	
	select 欄位名 from 資料來源 where 資料範圍 group by 欄位名 order by 欄位名 limit 限制條件;	# 指定輸出欄位名稱, *表示全部欄位 # 指定欄名+關係運算子(= > < <> ...), 可搭配 And Or # 升冪 ASC or 降冪 DESC # 加總、平均、計算筆數等
查詢多個欄位資料	select 欄位名, 欄位名, 欄位名 from 資料表名稱; SELECT id, name, preview_url, img_url, artist_id FROM tracks_album;	
比對字串	select 欄位名 from 資料表名稱 where 欄位名 like '%字串%'; - 前置樣板: 字串% - 中置樣板: %字串% - 後置樣板: %字串 SELECT * FROM tracks_album WHERE name LIKE 'music%';	
插入欄位資料	insert into 資料表名稱(欄位 1,欄位 2,欄位 3,欄位 4, 欄位 N) values('值 1', 值 2,'值 3','值 4', '值 N'); INSERT INTO tracks_album -> (id, name, preview_url, img_url, tracks_number, release_date, artist_id) -> VALUES -> ("123", "song", "123@com.tw", "123@com.tw", "1 ", "2017-05-28", "456 ");	
更新修改欄位資料 (一筆紀錄)	replace into 資料表名稱 (欄位 1,欄位 2,欄位 3,欄位 4, 欄位 N) values ('值 1','值 2','值 3','值 4', '值 N'); REPLACE INTO tracks_album -> (id, name) -> VALUES -> ('songid', 'namechange') replace into 跟 insert 功能類似, 不同點在於: replace into 首先嘗試插入資 料到表中, 1. 如果發現表中已經有此行資料(根據主鍵或者唯一索引判斷) 則先刪除此行資料, 然後插入新的資料。 2. 否則, 直接插入新資料。 要注意的是: 插入資料的表必須有主鍵或者是唯一索引! 否則的話, replace into 會直接插入資料, 這將導致表中出現重複的資料。	

更新修改欄位資料 (多筆紀錄)	update 資料表名稱 set 欄位 1='值 1',欄位 2='值 2',欄位 3='值 3',... 欄位 N='值 N' where 條件式 (例如 id='__' 或 name='____'); <code>UPDATE tracks_album</code> -> <code>SET name='song1'</code> -> <code>WHERE id=3;</code>
查看是否有空值	必須使用 IS NULL 和 IS NOT NULL (非使用 = 和 != 運算子) <code>SELECT * FROM tracks_album WHERE id IS NULL;</code>
資料表連接	<ul style="list-style-type: none"> - INNER JOIN (內連接): 獲取兩個表中欄位匹配關係的記錄 - LEFT JOIN (左連接): 獲取左表所有記錄, 即使右表沒有對應匹配的記錄 - RIGHT JOIN (右連接): 獲取右表所有記錄, 即使左表沒有對應匹配的記錄 <code>SELECT tableA.id, tableA.name, tableB.url</code> -> <code>FROM track_album tableA INNER JOIN track_features tableB</code> -> <code>ON tableA.name = tableB.name;</code>
複製資料表	<code>SHOW CREATE TABLE track_album \G;</code> <code>CREATE TABLE 'clone_track_album' ('id' int(11), 'name' varchar(30), PRIMARY KEY ('id'));</code> <code>INSERT INTO clone_track_album (id, name) ;</code> <code>SELECT id, name FROM track_album;</code> 另一種完整複製資料表的方法: <code>CREATE TABLE targetTable LIKE sourceTable;</code> <code>INSERT INTO targetTable SELECT * FROM sourceTable;</code>
單獨印出	\G 使每個欄位按行單獨列印 /G 的作用是將查到的結構旋轉 90 度變成縱向
計算資料筆數	<code>SELECT COUNT (*) FROM tracks_features;</code>
http://note.drx.tw/2012/12/mysql-syntax.html https://in.ncu.edu.tw/~center23/mysql/mysql_co.htm	

02 Django | 關聯 MySQL 數據庫

A. Django model 欄位類型清單

BooleanField(布林值)、TextField(文本)、DateField(日期)、FileField(文件)、
TimeField(時間)、EmailField(信箱)、FloatField(浮點數)、IntegerField(整數)、
IPAddressField(IP 位址)、PhoneNumberField(電話)、PositiveIntegerField(正整數)、
URLField(連結)、FilePathField(文件路徑)

<https://www.cnblogs.com/lhj588/archive/2012/05/24/2516040.html>

B. Django 配置

1. 初始化配置

- 安裝 pymysql

```
pip install pymysql
```

- 修改 setting.py

```
DATABASES = {
    'default': {
```

```
'ENGINE': 'django.db.backends.mysql',    # 資料庫引擎
'NAME': 'dbPersonMana',
'USER': 'root',
'PASSWORD': '',
'HOST': 'localhost',
'PORT': '3306',
```

```
}
```

藍框處可用成 my.cnf 檔案

<https://docs.djangoproject.com/en/2.2/ref/settings/#databases>

- 修改 __init__.py

```
import pymysql
pymysql.install_as_MySQLdb()
```

- makemigrations / migrate

```
python manage.py makemigrations appMana
python manage.py migrate
```

2. 創建 Model

- 編寫 models.py

```
# Create your models here.
class Album(models.Model):
    id = models.CharField(max_length=30, primary_key=True) # Album id
    name = models.CharField(max_length=100)
    artist = models.CharField(max_length=30)
    preview_url = models.URLField()
    img_url = models.URLField()
    tracks_number = models.IntegerField()
```

<https://www.jianshu.com/p/3b2c429ba509>

<https://www.cnblogs.com/lhj588/archive/2012/05/24/2516040.html>

03 實戰演練

A. 技巧

- Python3 pymysql 執行多條 sql 語句

```
# 事务处理
sql_1 = "UPDATE staff SET saving = saving + 1000 WHERE user_id = '1001' "
sql_2 = "UPDATE staff SET expend = expend + 1000 WHERE user_id = '1001' "
sql_3 = "UPDATE staff SET income = income + 2000 WHERE user_id = '1001' "

try:
    cursor.execute(sql_1) # 儲蓄增加1000
    cursor.execute(sql_2) # 支出增加1000
    cursor.execute(sql_3) # 收入增加2000
except Exception as e:
    connect.rollback() # 事务回滾
    print('事务处理失败', e)
else:
    connect.commit() # 事务提交
    print('事务处理成功', cursor.rowcount)
```

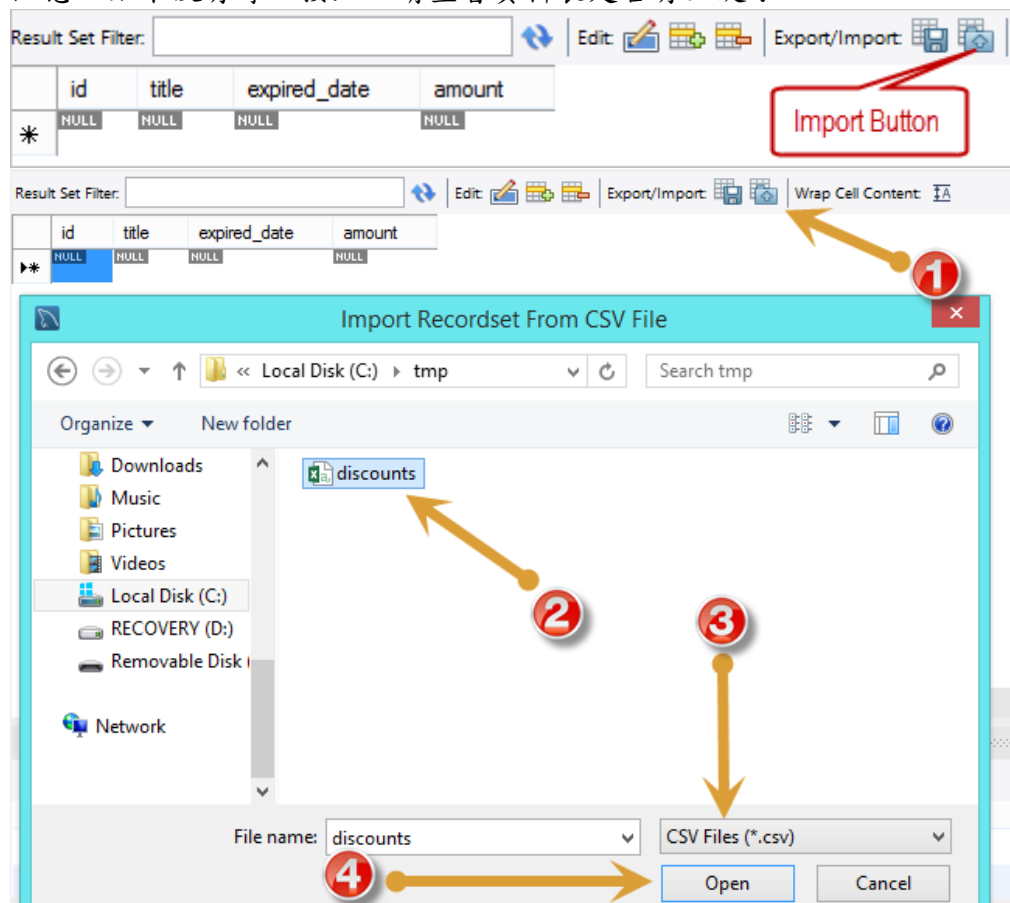
<https://blog.csdn.net/mengyangcust/article/details/84752342>

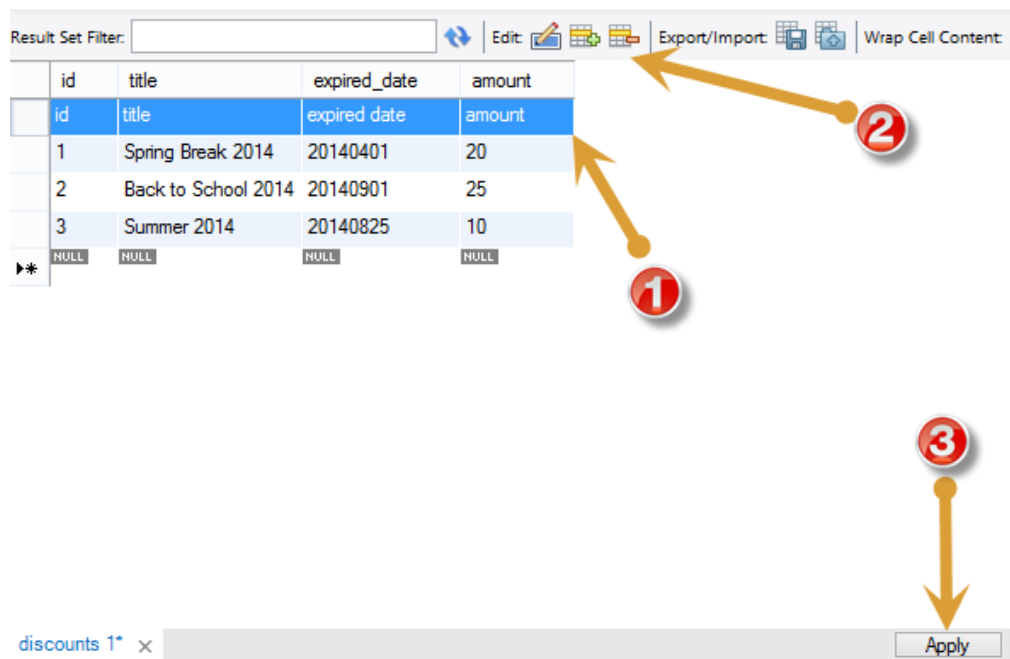
一 匯入 CSV

(1) MySQL workbench 方法

點擊匯入按鈕，選擇要匯入的檔，點擊 apply 匯入文件

注意：如果沒有導入按鈕，請查看資料表是否有主鍵！





(2) 下 SQL 語法方法

```
LOAD DATA LOCAL INFILE '/path/to/example.csv'
INTO TABLE tracks_audiolyricembedding (資料表名稱)
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

SELECT * FROM tracks_audiolyricembedding;
http://www.mysqltutorial.org/import-csv-file-mysql-table/
```

B. 程式碼

```
set SQL_SAFE_UPDATES = 0; # 解除輸入的安全模式
use tracks;
-- content base 推薦方法
-- 用戶 id
set @user_id = (select user from tracks_surveyresults);

-- 歌曲 ID track_id
set @id = '000xQL6tZNLJzIrtIgxqSI';

-- 專輯圖片 album_img_url
set @album_id = (select album_id from tracks_features where id = @id);
set @album_img_url = (select img_url from tracks_album where id = @album_id);

-- 歌手名稱 artist_name
set @artist_id = (select artist_id from tracks_features where id = @id);
```

```
set @artist_name = (select artist_name from tracks_artist where id = @artist_id);
```

-- 推薦方法 `recom_method`

查看 `surveyselect.py`

-- 推薦排序 `recom_rank`

-- 分數 `score`

```
set @recom_method = '0';
```

```
set @recom_rank = '1';
```

```
set @score = '5';
```

```
insert into tracks_recsysresults2 (user_id, album_img_url, track_id, artist_name,  
recom_method, recom_rank, score) values (@user_id, @album_img_url, @track_id,  
@artist_name, @recom_method, @recom_rank, @score);
```

```
select * from tracks_recsysresults2;
```

04 問題解決

A. 系統時間錯誤

MySQL 本身修改

1. 開啟 MySQL 8.0 Command Line Client (需輸入密碼：850121)
2. 查看自己的時區 (兩者保持與系統時間一致)

```
mysql> show variables like '%zone%';
```

```
mysql> select @@time_zone;
```

預設的是"UTC"，台灣是 UTC+8。

```
mysql> show variables like '%zone%';
```

```
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| system_time_zone |      |  
| time_zone       | SYSTEM |  
+-----+-----+  
2 rows in set, 1 warning (0.48 sec)
```

```
mysql> select now();
```

```
+-----+  
| now() |  
+-----+  
| 2019-06-19 19:26:25 |  
+-----+  
1 row in set (0.02 sec)
```

```
mysql> SELECT @@global.time_zone, @@session.time_zone;
```

```
+-----+-----+  
| @@global.time_zone | @@session.time_zone |  
+-----+-----+  
| SYSTEM            | SYSTEM              |  
+-----+-----+  
1 row in set (0.00 sec)
```

3. 修改檔案，`mysql\bin\my.ini`。
4. 到[mysqld]所在的位置，在它的下面加上 `default-time-zone = '+8:00'`
5. 存檔退出，重新啟動

<https://my.oschina.net/u/3672057/blog/2878383>

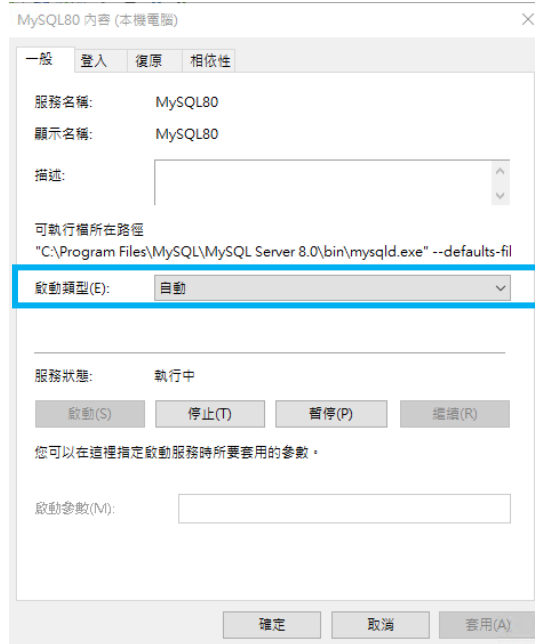
<https://s.yam.com/3XF3Y>

django 時區問題時間差 8 小時

- 如設置 USE_TZ=True，則儲存到資料庫中的時間是 UTC 時間(世界協調時間)。
- 修改 setting.py 檔案 註解掉 USE_TZ = True
<https://www.jianshu.com/p/c1dee7d3cbb9>

B. 解决 MySQL workbench 的 Can't connect to MySQL server on '127.0.0.1'(10061)問題

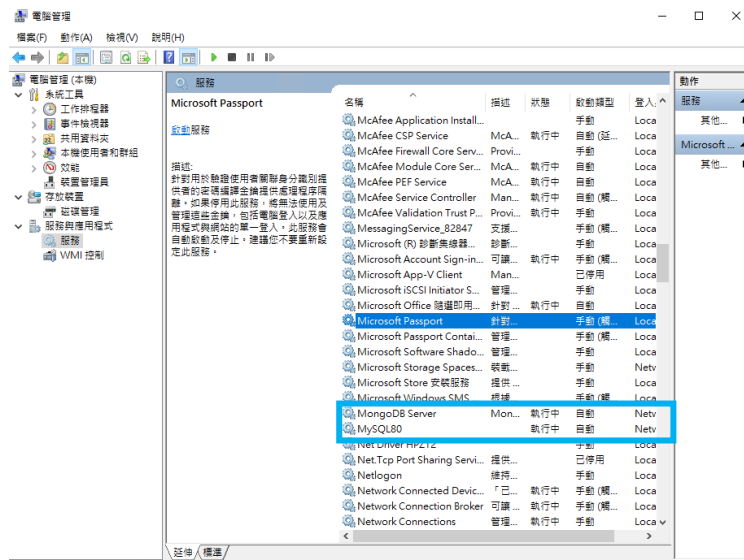
win + X 電腦管理 → 服務與應用程式 → 服務 → MySQL80 → 自動



<https://bbs.csdn.net/topics/390843506>

C. 解决 MongoDB Server 無法連接問題

win + X 電腦管理 → 服務與應用程式 → 服務 → MongoDB Sever → 自動



<https://bbs.csdn.net/topics/390843506>

D. 解除輸入的安全模式

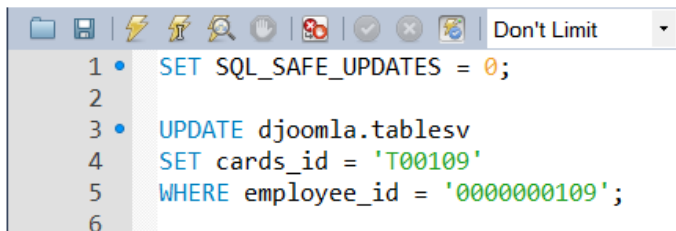
Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KEY column To disable safe mode, toggle the option in Preferences ->

SQL Queries and reconnect.

MySQL 運行在 safe-updates 模式下，該模式會導致非主鍵條件下無法執行 update 或者 delete 命令。

1. show variables like 'SQL_SAFE_UPDATES';查看開關狀態。
2. 執行命令 SET SQL_SAFE_UPDATES = 0;修改下資料庫模式

```
SET SQL_SAFE_UPDATES = 0;
```



The screenshot shows a MySQL command window with a toolbar at the top. The toolbar includes icons for file operations (new, open, save, print), editing (undo, redo, cut, copy, paste), and execution (run, stop, refresh). A dropdown menu on the right is set to 'Don't Limit'. The command window contains the following SQL queries:

```
1 • SET SQL_SAFE_UPDATES = 0;
2
3 • UPDATE djoomla.tablesv
4   SET cards_id = 'T00109'
5   WHERE employee_id = '0000000109';
6
```

<http://www.ucamc.com/e-learning/computer-skills/151-mysql-workbench-error-code-1175.html>

<https://chingsoo.pixnet.net/blog/post/292464274-%5Bmysql%5D-%E9%97%9C%E9%96%89safe-update-mode>

E. 更改 table 為何沒改變

數據表內容有更新，必須使用到該語句 commit 語法

```
mydb = pymysql.connect(host='127.0.0.1', port=3306, user='chiouchingyi',
passwd='850121', db='tracks')
mydb.commit()
```

Important!: Notice the statement: `mydb.commit()`. It is required to make the changes, otherwise no changes are made to the table.

https://www.w3schools.com/python/python_mysql_insert.asp