**目錄**

[Install Document 2](#_Toc12747998)

[01 複製既有的遠端數據庫 2](#_Toc12747999)

[02 查看遠端數據庫分支的修改內容 2](#_Toc12748000)

[03 合併分支 2](#_Toc12748001)

[04 更新本機上檔案 2](#_Toc12748002)

[05 位址轉到專案 3](#_Toc12748003)

[06 修改環境參數 3](#_Toc12748004)

[07 Model Migration 3](#_Toc12748005)

[08 執行伺服器 5](#_Toc12748006)

[09 申請MySQL / ngrok 5](#_Toc12748007)

[10 執行ngrok 5](#_Toc12748008)

[11 新增使用者Creating users 6](#_Toc12748009)

[Project Document 7](#_Toc12748010)

[01 GitHub 7](#_Toc12748011)

[02 顯示所有的Table列表 7](#_Toc12748012)

[03 資料集 7](#_Toc12748013)

[04 使用Table 10](#_Toc12748014)

[05 查看特定Table 10](#_Toc12748015)

[06 新增Table 10](#_Toc12748016)

[07 推薦清單資料庫欄位 10](#_Toc12748017)

[08 更改CODE 10](#_Toc12748018)

[MySQL Document 12](#_Toc12748019)

[01 SQL語法 12](#_Toc12748020)

[02 Django | 關聯MySql數據庫 13](#_Toc12748021)

[03 實戰演練 14](#_Toc12748022)

[04 問題解決 17](#_Toc12748023)

# Install Document

## 複製既有的遠端數據庫

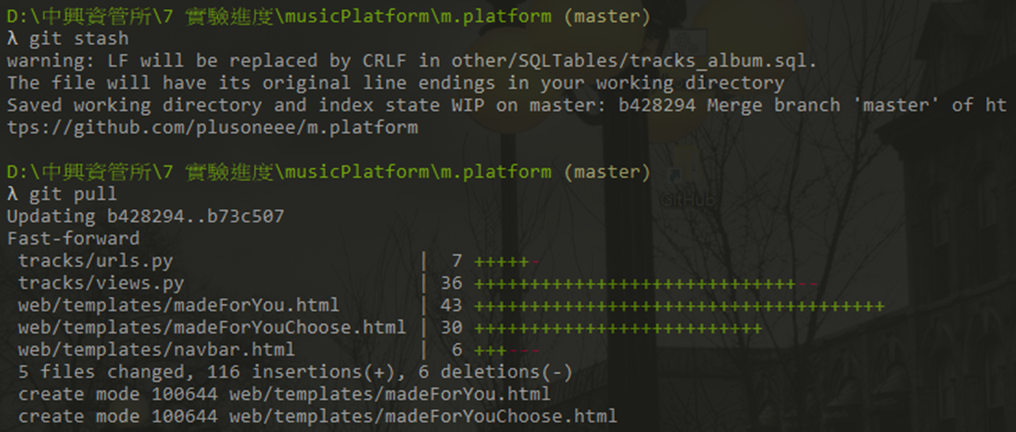
λ git clone <url>

## 查看遠端數據庫分支的修改內容

λ git fetch <repository>

## 合併分支

λ git stash

****

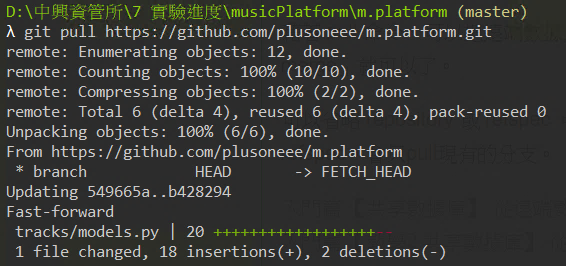
## 更新本機上檔案

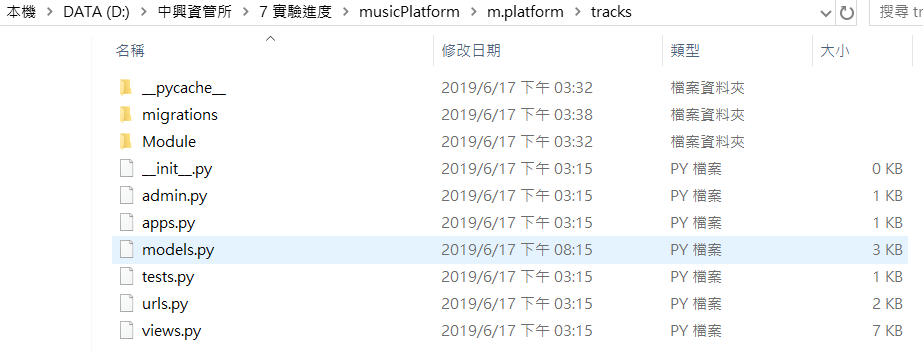
λ git pull <repository網址>

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





## 位址轉到專案

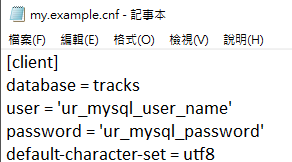
移到D槽 → λ d:

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

## 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



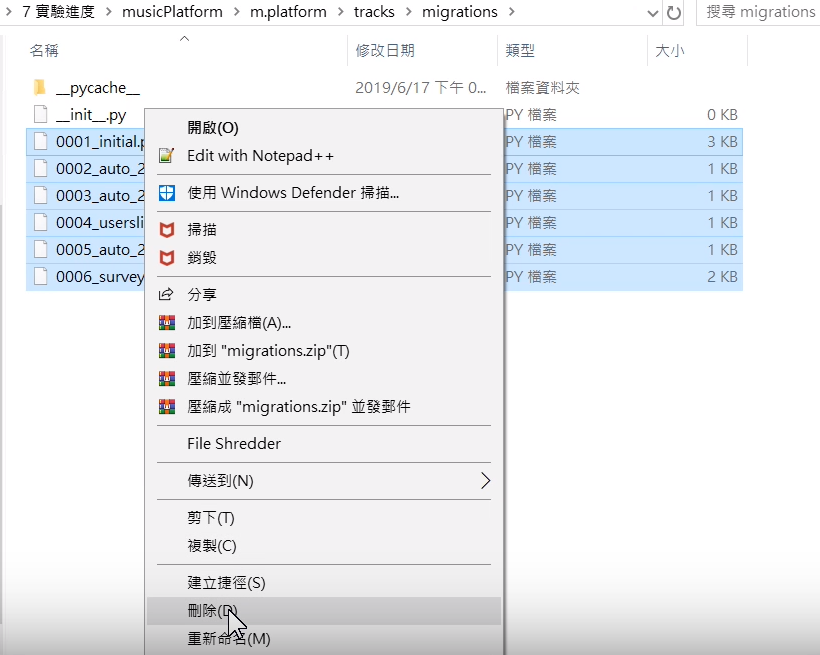
## Model Migration

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

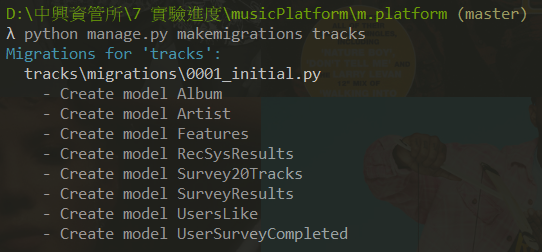
移至 D:\中興資管所\7 實驗進度\musicPlatform\m.platform\tracks\migrations

刪除底下0001\_initial.py …等檔案

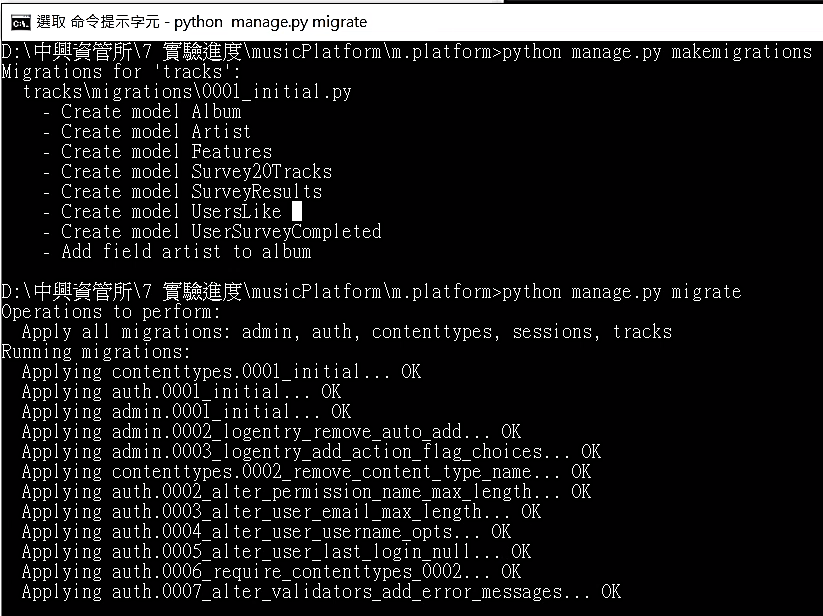
留下「\_\_pycache\_\_資料夾」和「\_\_init\_\_.py檔案」



正確結果：



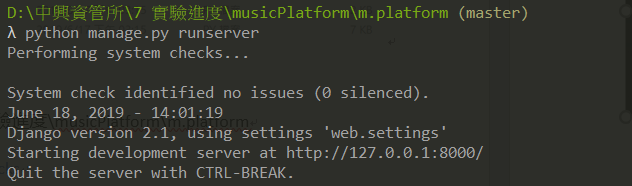
λ python manage.py migrate



## 執行伺服器

λ python manage.py runserver

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



## 申請MySQL / ngrok

MySQL

帳號：root / chiouchingyi (擇一)

密碼：850121

ngrok

帳號：chiouchingyi0121@gmail.com

密碼：jenny850121

## 執行ngrok

1. 點擊ngrok.exe執行檔

D:\程式檔\ ngrok.exe

1. 連接至帳號Connect your acount (到ngrok網站Auth查看)

Your Tunnel Authtoken : 7v3SqyXN25gZEtqSKrV9H\_4hphM6C4bNgyWCFCt5RG4

1. 執行檔執行指令

λ ngrok authtoken 7v3SqyXN25gZEtqSKrV9H\_4hphM6C4bNgyWCFCt5RG4

1. 執行HTTP tunnel在自己的port

λ ngrok.exe http 8000 (藍框處)



1. 修改setting. py檔案

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

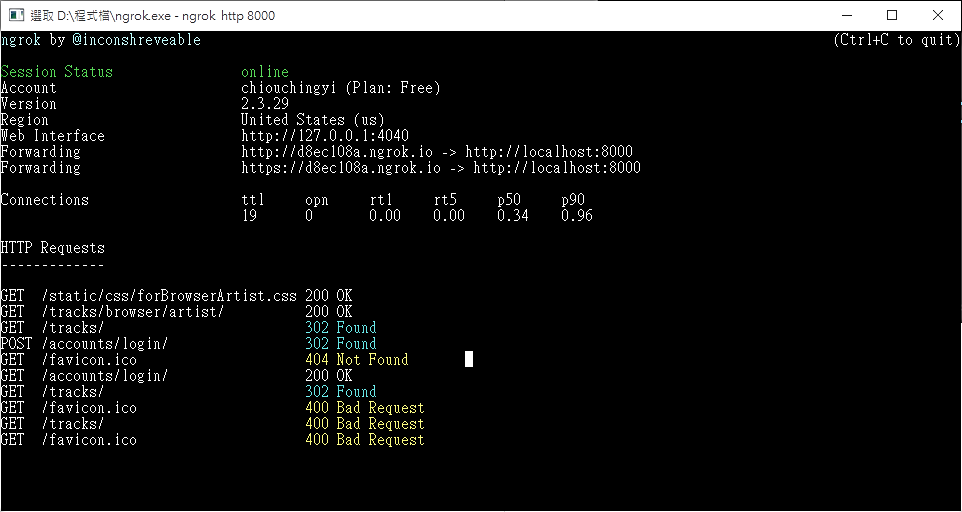
ALLOWED\_HOSTS = [] 改為 ALLOWED\_HOSTS = ['\*']

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

1. 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)

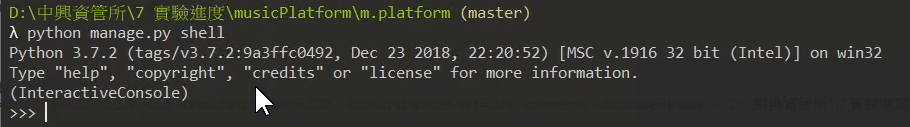


## 新增使用者Creating users

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

1. Cmder執行manage.py檔

λ python manage.py shell



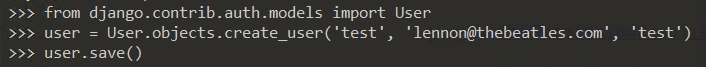
1. 輸入create指令

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()



1. 重新啟動伺服器

λ python manage.py runserver



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

# Project Document

## GitHub

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

## 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\_audiolyricsembedding | 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 |

## 資料集

1. **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 | | | |

1. **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 | | | | | | |

1. **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 |

1. **音頻歌詞向量資料集**

|  |  |  |  |
| --- | --- | --- | --- |
| Table名稱：tracks\_audiolyricsembedding | | | |
| id | name | audio\_vector | lyrics\_vector |
| 歌曲ID | 歌曲名稱 | 音頻向量 | 音頻向量 |

1. **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 |

1. **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 |

1. **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 |

1. **調查歌曲特徵資料集**

|  |  |  |  |
| --- | --- | --- | --- |
| tracks\_survey20tracks\_vector(edit).csv | | | |
| Table名稱：tracks\_survey20tracksfeatures | | | |
| id | name | audio\_vecter | lyrics\_vecter |
| 歌曲ID | 歌曲名稱 | 音頻向量 | 歌詞向量 |

1. **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 |

1. **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 | | | | | | |

1. **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 |

1. **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 |

1. **調查完成**

|  |  |  |
| --- | --- | --- |
| Table名稱：tracks\_usersurveycompleted | | |
| id | user | complete\_date |
| 編號 | 用戶ID | 完成時間 |

1. **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

## 使用Table

use tracks;

## 查看特定Table

select \* from tracks\_artist;

select \* from tracks\_features;

select \* from tracks\_survey20tracks;

select \* from auth\_user;

## 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 ()

## 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

## 推薦方法

1. 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'

## 更改CODE

|  |  |
| --- | --- |
| 原始 | 更改 |
| ALLOWED\_HOSTS = [] | ALLOWED\_HOSTS = ['\*'] |
| USE\_TZ = True | ##USE\_TZ = True (註解掉) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# MySQL Document

## 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   1. INT (整數) 2. CHAR (1~255字元字串) 3. VARCHAR (不超過255字元不定長度字串) 4. 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](mailto:123@com.tw)", "[123@com.tw](mailto: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='\_\_\_\_' );  UPDATE tracks\_album  -> SET name='song1'  -> WHERE id=3; | | |
| 查看是否有空值 | 必須使用 IS NULL和IS NOT NULL (非使用 = 和 != 運算子)  SELECT \* FROM tracks\_album WHERE id IS NULL; | | |
| 資料表連接 | * 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   SELECT tableA.id, tableA.name, tableB.url  -> FROM track\_album tableA INNER JOIN track\_features tableB  -> ON tableA.name = tableB.name; | | |
| 複製資料表 | SHOW CREATE TABLE track\_album \G;  CREATE TABLE 'clone\_ track\_album' ('id' int(11), 'name' varchar(30), PRIMARY KEY ('id'));  INSERT INTO clone\_ track\_album (id, name) ;  SELECT id, name FROM track\_album;  另一種完整複製資料表的方法:  CREATE TABLE targetTable LIKE sourceTable;  INSERT INTO targetTable SELECT \* FROM sourceTable; | | |
| 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 | SELECT COUNT (\*) FROM tracks\_features; | | |
| http://note.drx.tw/2012/12/mysql-syntax.html  https://in.ncu.edu.tw/~center23/mysql/mysql\_co.htm | | | |

## Django | 關聯MySql數據庫

1. **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>

1. **Django配置**
2. 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

1. 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()

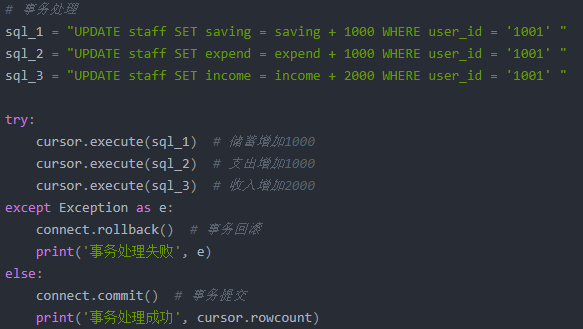
tracks\_number = models.IntegerField()

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

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

## 實戰演練

1. 技巧
   * Python3 pymysql執行多條sql語句



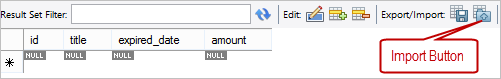
<https://blog.csdn.net/mengyangcust/article/details/84752342>

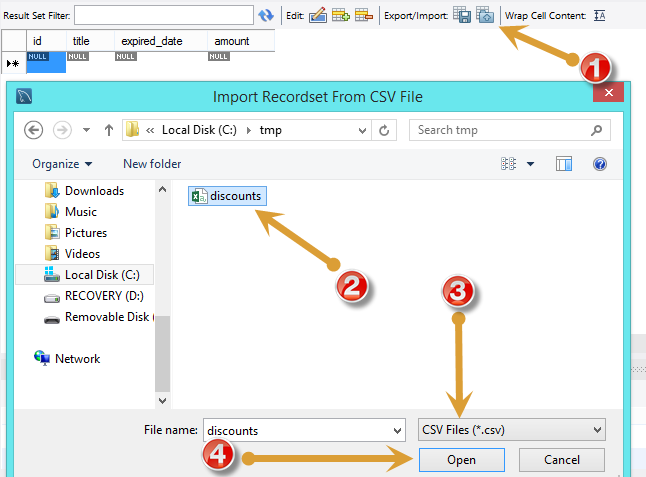
* + 匯入csv

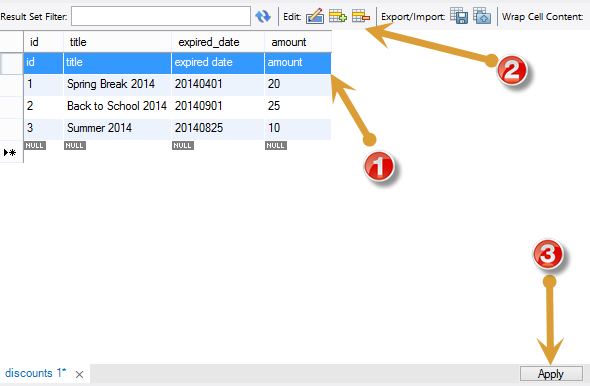
1. MySQL workbench方法

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！







1. 下SQL語法方法

LOAD DATA LOCAL INFILE '/path/to/example.csv'

INTO TABLE tracks\_audiolyricsembedding (資料表名稱)

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

SELECT \* FROM tracks\_audiolyricsembedding;

<http://www.mysqltutorial.org/import-csv-file-mysql-table/>

1. 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 = '000xQL6tZNLJzIrtIgxqSl';

-- 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;

## 問題解決

1. **系統時間錯誤**

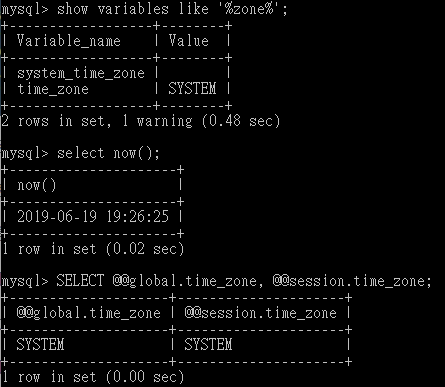
MySQL本身修改

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。



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

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

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

------------------------------------------------------------------------------------------------------------------------------

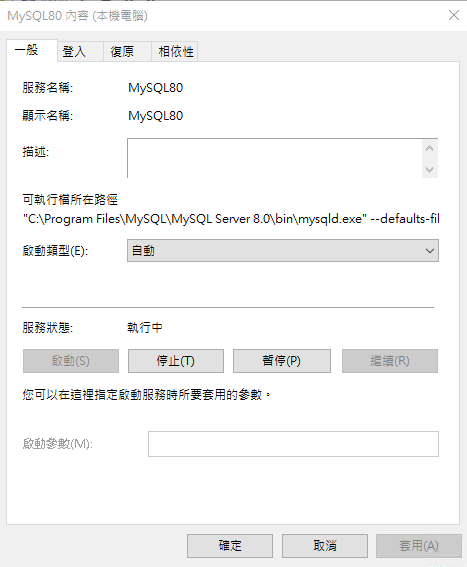
django時區問題時間差8小時

* 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>

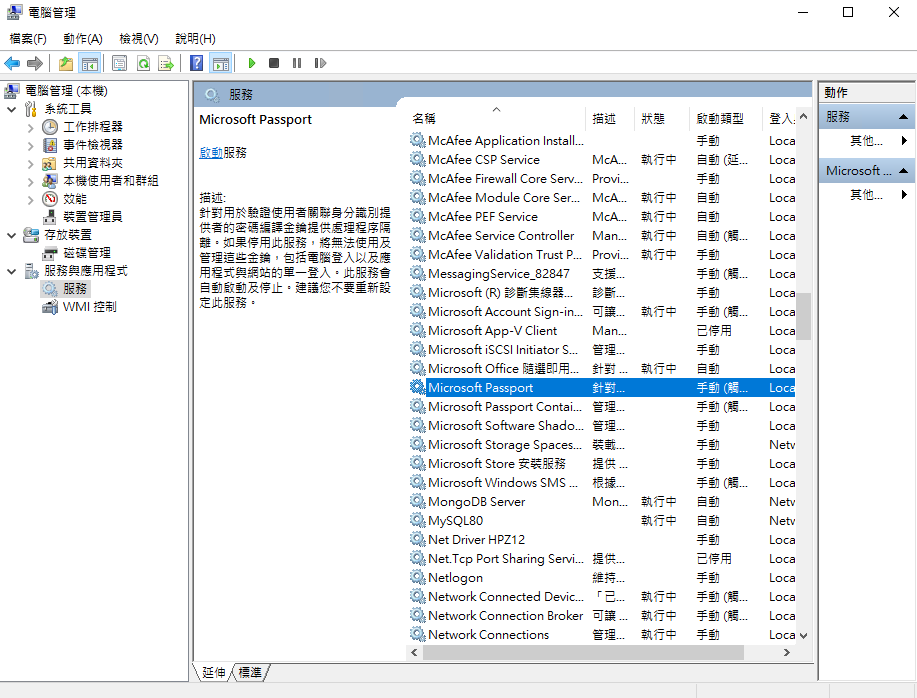
1. **解决MySQL workbench的Can't connect to MySQL server on '127.0.0.1'(10061)問題**

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

 [https://bbs. csdn.net/topics/390843506](https://bbs.csdn.net/topics/390843506)

1. **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



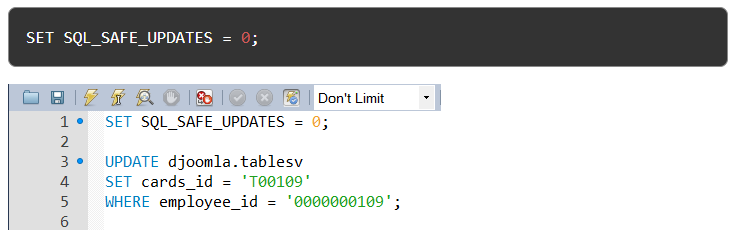
<https://bbs.csdn.net/topics/390843506>

1. **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



<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>

1. **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>