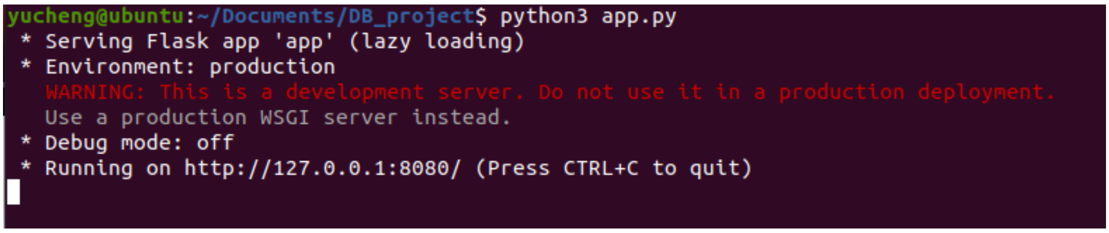
Database Application Document

Motivations

動漫是日本相當重要的文化，其影響力遍及全世界。相信大多數人都有過看動畫的經驗。從 1980 年代至今，全世界有成千上萬部動畫。常常我們閒暇之餘想找一部動畫來看，卻又沒什麼目標，相當苦惱。因此，我想到可以製作一個推薦動畫的應用程式，選取年代範圍後，可以根據知名度或網友評分來排序並列出表格。列出的資料筆數也是可以控制的。這樣就算沒有目標，也可以找到當代評分高的、知名度高的動畫來看。另外還要設計一個附屬功能，就是輸入動漫名稱，就能查詢到它的故事摘要。因為要是只看知名度或評分，沒辦法知道這部動畫的內容符不符合使用者的興趣。

Application description

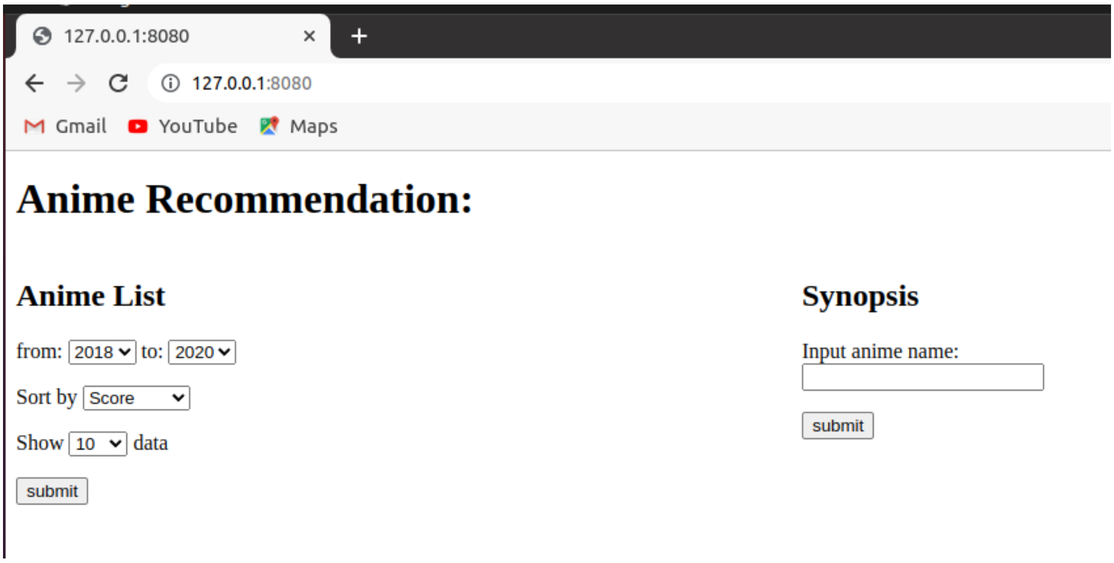
我的 App 是一個運行在本地端 port 8080 的網頁 (http://127.0.0.1:8080/)。我使用 html5 來實作網頁的前端、python 的 flask 套件實作網頁後端及連接 AWS 的資料庫。在輸入網址前要先運行後端程式:



```
yucheng@ubuntu:~/Documents/DB_project$ python3 app.py
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
```

接著就可以在瀏覽器 (建議使用 Chrome) 輸入網址

(http://127.0.0.1:8080/) 來進入網頁 App。接著移動到下方功能區:



127.0.0.1:8080

127.0.0.1:8080

Gmail YouTube Maps

Anime Recommendation:

Anime List

from: 2018 to: 2020

Sort by Score

Show 10 data

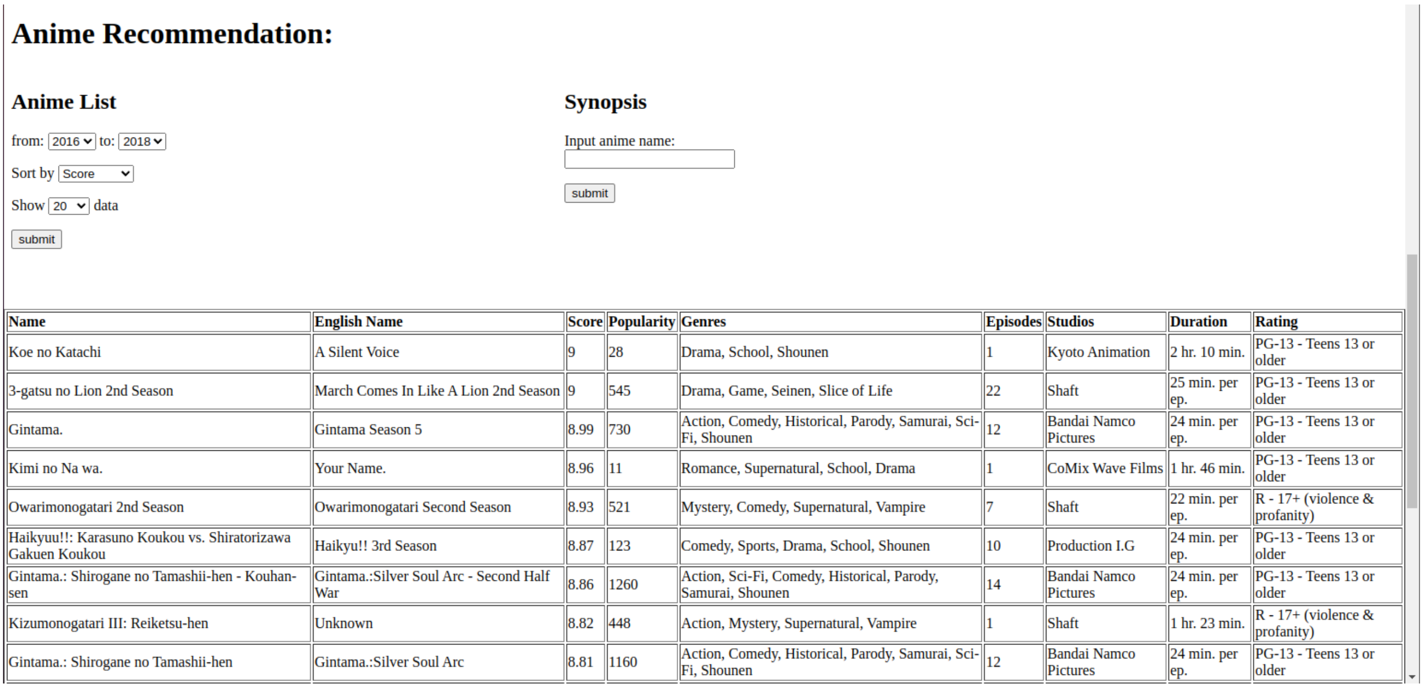
submit

Synopsis

Input anime name:

submit

可以看到左邊的 anime list 功能及右邊的 get synopsis 功能。可以使用下壓式選單選取動畫的年份範圍(1980~2021)、根據網友評分(Score)或網路人氣(Popularity)列表、及要列出前幾筆資料(10, 20, 50, 100)。做好選擇後按下 submit 鍵，以下圖為例:



Anime Recommendation:

Anime List

from: 2016 to: 2018

Sort by Score

Show 20 data

submit

Synopsis

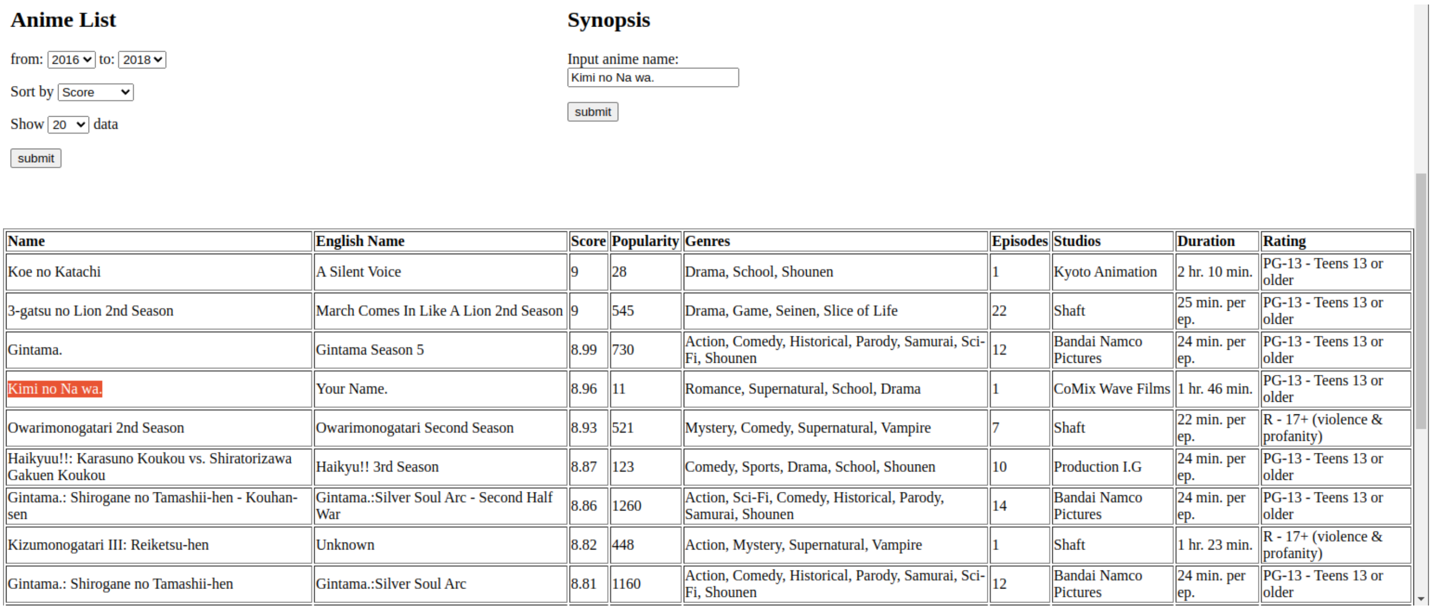
Input anime name:

submit

Name	English Name	Score	Popularity	Genres	Episodes	Studios	Duration	Rating
Koe no Katachi	A Silent Voice	9	28	Drama, School, Shounen	1	Kyoto Animation	2 hr. 10 min.	PG-13 - Teens 13 or older
3-gatsu no Lion 2nd Season	March Comes In Like A Lion 2nd Season	9	545	Drama, Game, Seinen, Slice of Life	22	Shaft	25 min. per ep.	PG-13 - Teens 13 or older
Gintama.	Gintama Season 5	8.99	730	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	12	Bandai Namco Pictures	24 min. per ep.	PG-13 - Teens 13 or older
Kimi no Na wa.	Your Name.	8.96	11	Romance, Supernatural, School, Drama	1	CoMix Wave Films	1 hr. 46 min.	PG-13 - Teens 13 or older
Owarimonogatari 2nd Season	Owarimonogatari Second Season	8.93	521	Mystery, Comedy, Supernatural, Vampire	7	Shaft	22 min. per ep.	R - 17+ (violence & profanity)
Haikyuu!!: Karasuno Koukou vs. Shiratorizawa Gakuen Koukou	Haikyu!! 3rd Season	8.87	123	Comedy, Sports, Drama, School, Shounen	10	Production I.G	24 min. per ep.	PG-13 - Teens 13 or older
Gintama.: Shirogane no Tamashii-hen - Kouhan-sen	Gintama.:Silver Soul Arc - Second Half War	8.86	1260	Action, Sci-Fi, Comedy, Historical, Parody, Samurai, Shounen	14	Bandai Namco Pictures	24 min. per ep.	PG-13 - Teens 13 or older
Kizumonogatari III: Reiketsu-hen	Unknown	8.82	448	Action, Mystery, Supernatural, Vampire	1	Shaft	1 hr. 23 min.	R - 17+ (violence & profanity)
Gintama.: Shirogane no Tamashii-hen	Gintama.:Silver Soul Arc	8.81	1160	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	12	Bandai Namco Pictures	24 min. per ep.	PG-13 - Teens 13 or older

我選擇列出年份 2016 至 2018 年、根據 Score 排序的前 20 名動畫。

按下 submit 後可以得到如上圖之列表。表中呈現了每部動畫的動畫名、動畫英文名、分數、人氣、類別、集數、製作公司、(單集)長度、年齡分級這些資訊供使用者參考。若使用者看到有興趣的動畫，可以將動畫名稱 (Name) 複製，貼到右邊 Synopsis 功能。現在假設我想知道「你的名字」的內容摘要，如下圖所示:



Anime List

from: 2016 to: 2018

Sort by Score

Show 20 data

submit

Synopsis

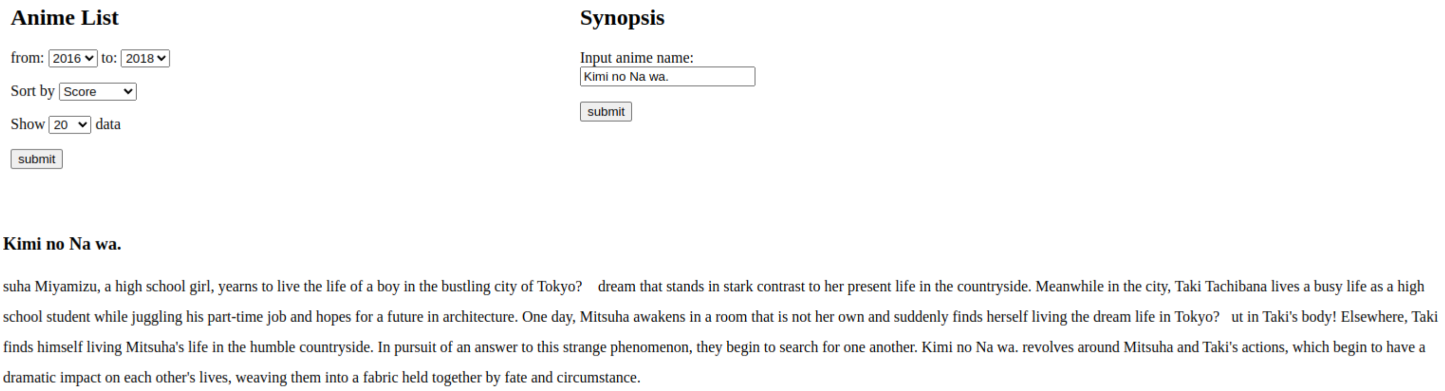
Input anime name:

Kimi no Na wa.

submit

Name	English Name	Score	Popularity	Genres	Episodes	Studios	Duration	Rating
Koe no Katachi	A Silent Voice	9	28	Drama, School, Shounen	1	Kyoto Animation	2 hr. 10 min.	PG-13 - Teens 13 or older
3-gatsu no Lion 2nd Season	March Comes In Like A Lion 2nd Season	9	545	Drama, Game, Seinen, Slice of Life	22	Shaft	25 min. per ep.	PG-13 - Teens 13 or older
Gintama.	Gintama Season 5	8.99	730	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	12	Bandai Namco Pictures	24 min. per ep.	PG-13 - Teens 13 or older
Kimi no Na wa.	Your Name.	8.96	11	Romance, Supernatural, School, Drama	1	CoMix Wave Films	1 hr. 46 min.	PG-13 - Teens 13 or older
Owarimonogatari 2nd Season	Owarimonogatari Second Season	8.93	521	Mystery, Comedy, Supernatural, Vampire	7	Shaft	22 min. per ep.	R - 17+ (violence & profanity)
Haikyuu!!: Karasuno Koukou vs. Shiratorizawa Gakuen Koukou	Haikyu!! 3rd Season	8.87	123	Comedy, Sports, Drama, School, Shounen	10	Production I.G	24 min. per ep.	PG-13 - Teens 13 or older
Gintama.: Shirogane no Tamashii-hen - Kouhan-sen	Gintama.:Silver Soul Arc - Second Half War	8.86	1260	Action, Sci-Fi, Comedy, Historical, Parody, Samurai, Shounen	14	Bandai Namco Pictures	24 min. per ep.	PG-13 - Teens 13 or older
Kizumonogatari III: Reiketsu-hen	Unknown	8.82	448	Action, Mystery, Supernatural, Vampire	1	Shaft	1 hr. 23 min.	R - 17+ (violence & profanity)
Gintama.: Shirogane no Tamashii-hen	Gintama.:Silver Soul Arc	8.81	1160	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	12	Bandai Namco Pictures	24 min. per ep.	PG-13 - Teens 13 or older

將動畫名稱複製、在右方欄位貼上並按下右邊 submit 鍵:



Anime List

from: 2016 to: 2018

Sort by Score

Show 20 data

submit

Synopsis

Input anime name:

Kimi no Na wa.

submit

Kimi no Na wa.

suha Miyamizu, a high school girl, yearns to live the life of a boy in the bustling city of Tokyo? dream that stands in stark contrast to her present life in the countryside. Meanwhile in the city, Taki Tachibana lives a busy life as a high school student while juggling his part-time job and hopes for a future in architecture. One day, Mitsuha awakens in a room that is not her own and suddenly finds herself living the dream life in Tokyo? ut in Taki's body! Elsewhere, Taki finds himself living Mitsuha's life in the humble countryside. In pursuit of an answer to this strange phenomenon, they begin to search for one another. Kimi no Na wa. revolves around Mitsuha and Taki's actions, which begin to have a dramatic impact on each other's lives, weaving them into a fabric held together by fate and circumstance.

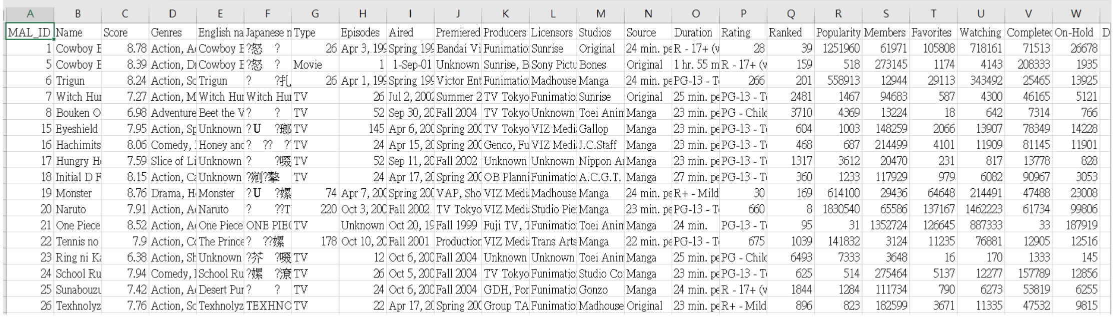
就可以得到該動畫的內容摘要。

Data sources and import

我的 data 是從 Kaggle 上找來的。連結如下:

https://www.kaggle.com/datasets/hernan4444/anime-recommendation-database-2020

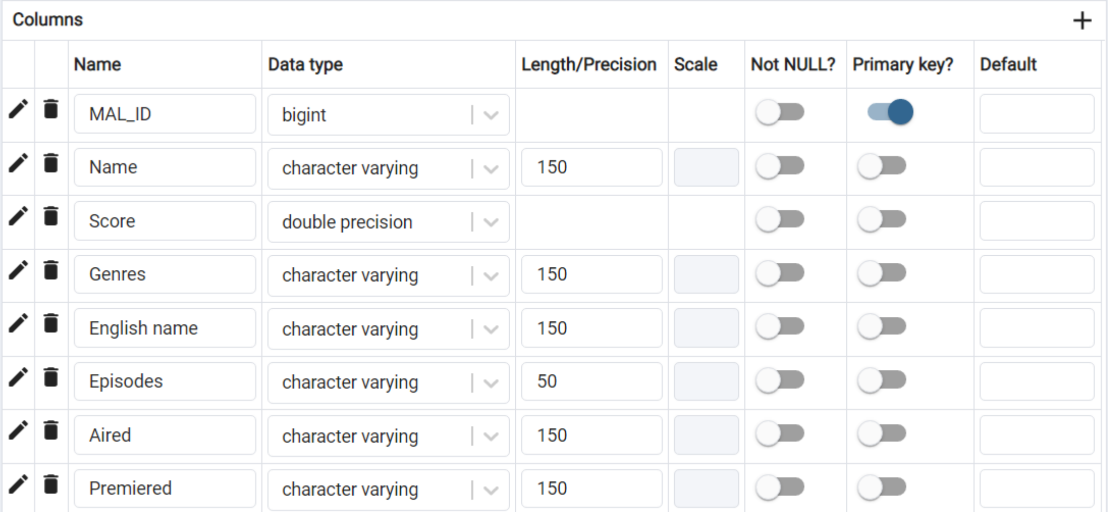
這個 repository 中有五個 csv 檔，取其中的兩個來做使用。這些 data 剛下載下來是十分凌亂的，許多欄位是空的，甚至有些欄位根本沒有對齊，如下所示:



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
MAL_ID	Name	Score	Genres	English na	Japanese n	Type	Episodes	Aired	Premiered	Producers	Licensors	Studios	Source	Duration	Rating	Ranked	Popularity	Members	Favorites	Watching	Completed	On-Hold	D
1	Cowboy E	8.78	Action, A	Cowboy E	?怒 ?	26	Apr 3, 199	Spring 199	Bandai Vi	Funimatio	Sunrise	Original	24 min. pe	R - 17+ (v	28	39	1251960	61971	105808	718161	71513	26678	
5	Cowboy E	8.39	Action, Dr	Cowboy E	?怒 ?	Movie	1	1-Sep-01	Unknown	Sunrise, B	Sony Pictu	Bones	Original	1 hr. 55 m	R - 17+ (v	159	518	273145	1174	4143	208333	1935	
6	Trigun	8.24	Action, Sc	Trigun	? ?扎	26	Apr 1, 199	Spring 199	Victor Ent	Funimatio	Madhouse	Manga	24 min. pe	PG-13 - T	266	201	558913	12944	29113	343492	25465	13925	
7	Witch Hu	7.27	Action, M	Witch Hu	Witch Hu	TV	26	Jul 2, 2002	Summer 2	TV Tokyo	Funimatio	Sunrise	Original	25 min. pe	PG-13 - T	2481	1467	94683	587	4300	46165	5121	
8	Bouken O	6.98	Adventure	Beet the V	? ?	TV	52	Sep 30, 20	Fall 2004	TV Tokyo	Unknown	Toei Anim	Manga	23 min. pe	PG - Child	3710	4369	13224	18	642	7314	766	
15	Eyeshield	7.95	Action, Sp	Unknown	?U ?鄉	TV	145	Apr 6, 200	Spring 200	TV Tokyo	VIZ Medi	Gallop	Manga	23 min. pe	PG-13 - T	604	1003	148259	2066	13907	78349	14228	
16	Hachimits	8.06	Comedy, I	Honey and	? ?? ?	TV	24	Apr 15, 20	Spring 200	Genco, Fu	VIZ Medi	J.C.Staff	Manga	23 min. pe	PG-13 - T	468	687	214499	4101	11909	81145	11901	
17	Hungry He	7.59	Slice of Li	Unknown	? ?啜	TV	52	Sep 11, 20	Fall 2002	Unknown	Unknown	Nippon A	Manga	23 min. pe	PG-13 - T	1317	3612	20470	231	817	13778	828	
18	Initial D F	8.15	Action, Ca	Unknown	?剜?擎	TV	24	Apr 17, 20	Spring 200	OB Planni	Funimatio	A.C.G.T.	Manga	27 min. pe	PG-13 - T	360	1233	117929	979	6082	90967	3053	
19	Monster	8.76	Drama, H	Monster	?U ?嫘	74	Apr 7, 200	Spring 200	VAP, Sho	VIZ Medi	Madhouse	Manga	24 min. pe	R+ - Mild	30	169	614100	29436	64648	214491	47488	23008	
20	Naruto	7.91	Action, A	Naruto	? ??T	220	Oct 3, 200	Fall 2002	TV Tokyo	VIZ Medi	Studio Pie	Manga	23 min. pe	PG-13 - T	660	8	1830540	65586	137167	1462223	61734	99806	
21	One Piece	8.52	Action, A	One Piece	ONE PIE(	TV	Unknown	Oct 20, 19	Fall 1999	Fuji TV, T	Funimatio	Toei Anim	Manga	24 min.	PG-13 - T	95	31	1352724	126645	887333	33	187919	
22	Tennis no	7.9	Action, C	The Prince	? ??嫘	178	Oct 10, 20	Fall 2001	Production	VIZ Medi	Trans Arts	Manga	22 min. pe	PG-13 - T	675	1039	141832	3124	11235	76881	12905	12516	
23	Ring ni Ka	6.38	Action, Sh	Unknown	?芥 ?啜	TV	12	Oct 6, 200	Fall 2004	Unknown	Unknown	Toei Anim	Manga	25 min. pe	PG - Child	6493	7333	3648	16	170	1333	145	
24	School Ru	7.94	Comedy, I	School Ru	?嫘 ?潦	TV	26	Oct 5, 200	Fall 2004	TV Tokyo	Funimatio	Studio Co	Manga	23 min. pe	PG-13 - T	625	514	275464	5137	12277	157789	12856	
25	Sunabouzu	7.42	Action, A	Desert Pur	? ?	TV	24	Oct 6, 200	Fall 2004	GDH, Por	Funimatio	Gonzo	Manga	24 min. pe	R - 17+ (v	1844	1284	111734	790	6273	53819	6255	
26	Texhnolyz	7.76	Action, Sc	Texhnolyz	TEXHNO	TV	22	Apr 17, 20	Spring 200	Group TA	Funimatio	Madhouse	Original	23 min. pe	R+ - Mild	896	823	182599	3671	11335	47532	9815	

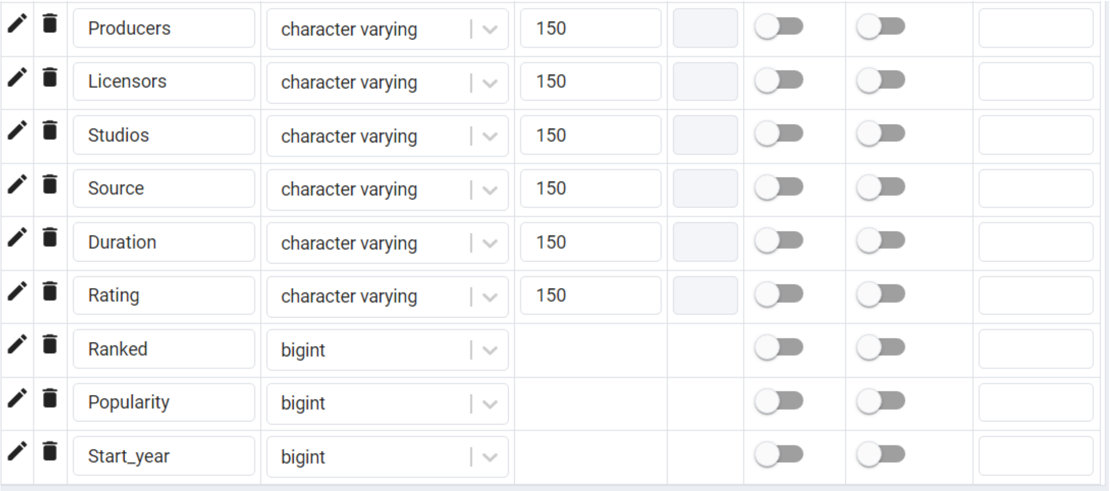
從上圖中可以很明顯的知道，這些檔案連 1NF 都不符合。經過我的觀察及整理後，將兩個 preprocessing 過檔案透過 pgAdmin4 裡的 import 功能匯入。

table animelist:



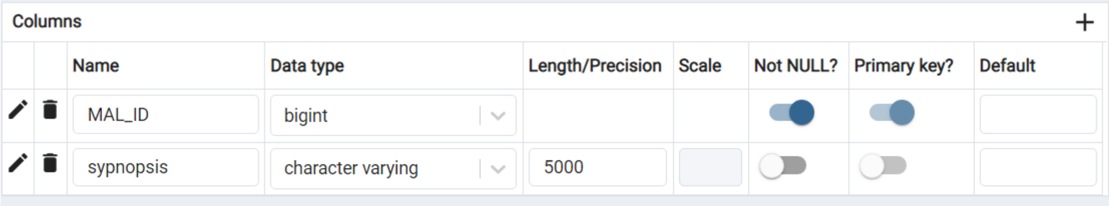
Columns +

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
MAL_ID	bigint			○	●	
Name	character varying	150		○	○	
Score	double precision			○	○	
Genres	character varying	150		○	○	
English name	character varying	150		○	○	
Episodes	character varying	50		○	○	
Aired	character varying	150		○	○	
Premiered	character varying	150		○	○	



Producers	character varying	150		○	○	
Licensors	character varying	150		○	○	
Studios	character varying	150		○	○	
Source	character varying	150		○	○	
Duration	character varying	150		○	○	
Rating	character varying	150		○	○	
Ranked	bigint			○	○	
Popularity	bigint			○	○	
Start_year	bigint			○	○	

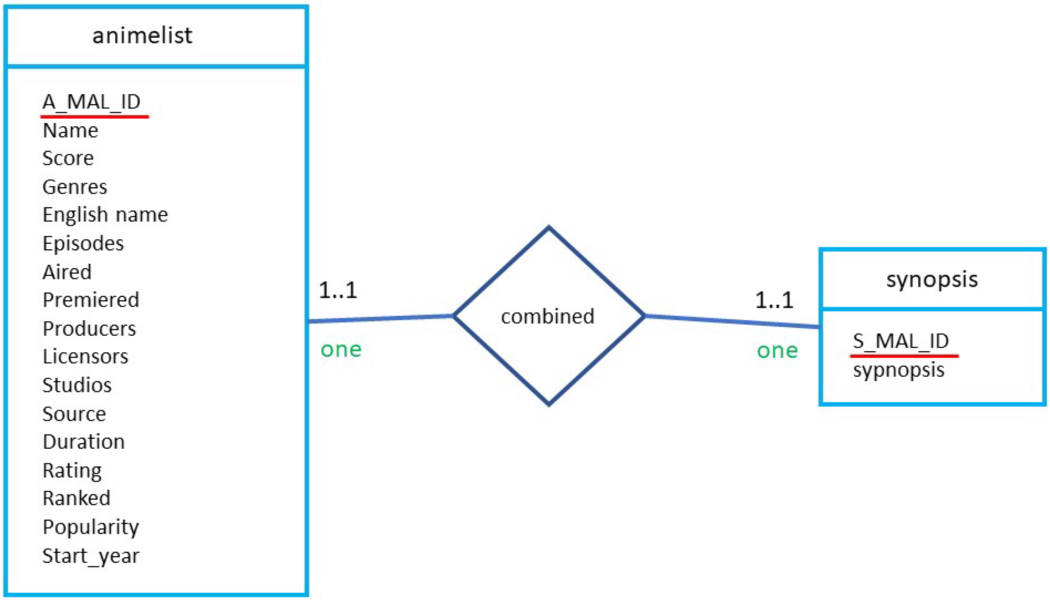
table synopsis:



Columns +

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
MAL_ID	bigint			●	●	
sypnopsis	character varying	5000		○	○	

Database schema

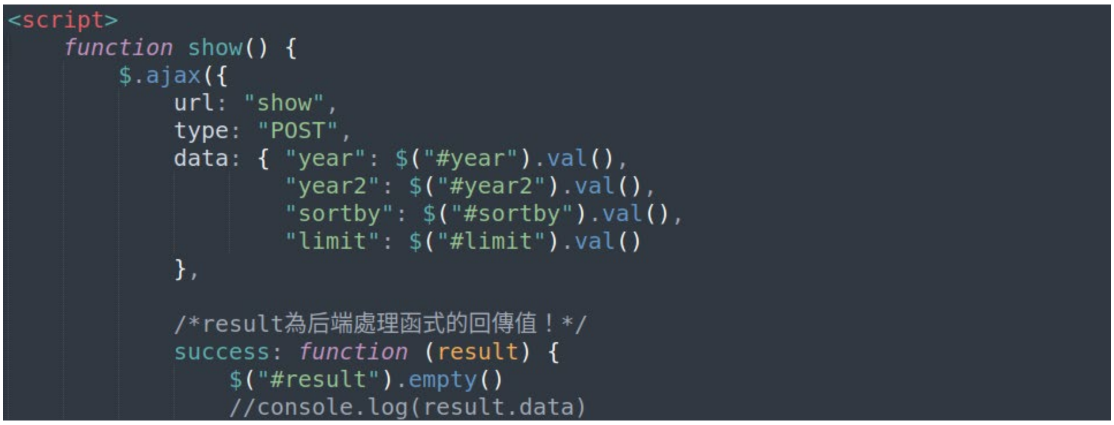


我用兩個 table，分別是 animelist 和 synopsis。他們屬於 one to one relation。animelist 的 primary key 是 A_MAL_ID、synopsis 的

primary key 是 S_MAL_ID。這兩者又互為 foreign key。畫成 ER diagram 時為了區分兩個 ID 欄位，將它們加上了各自的 prefix。實際寫成 table 時，還是都用 MAL_ID 命名。兩個 table 都只有單一個 primary key，不會有 partial dependencies 問題，符合 2NF。至於 3NF 可能會有一些 transitive dependencies 成立，例如 Name determines English name。但我們每次輸出列表都要包含這些欄位，如果每次查找都需要為了一兩個欄位而做 join，會有成本太高和效率低落的問題。因此，我選擇不對 animelist 做分割。另一方面，對於 synopsis 來說，我認為取得作品摘要屬於附屬的功能，使用上不會有 list 這麼頻繁。如果將兩者一開始就 join 成一張大 table，那麼 synopsis 欄位將會很常被棄用，而它又儲存了大量文字資料，這明顯是沒有效率的。因此我選擇在執行摘要功能時再做 join 來取得。

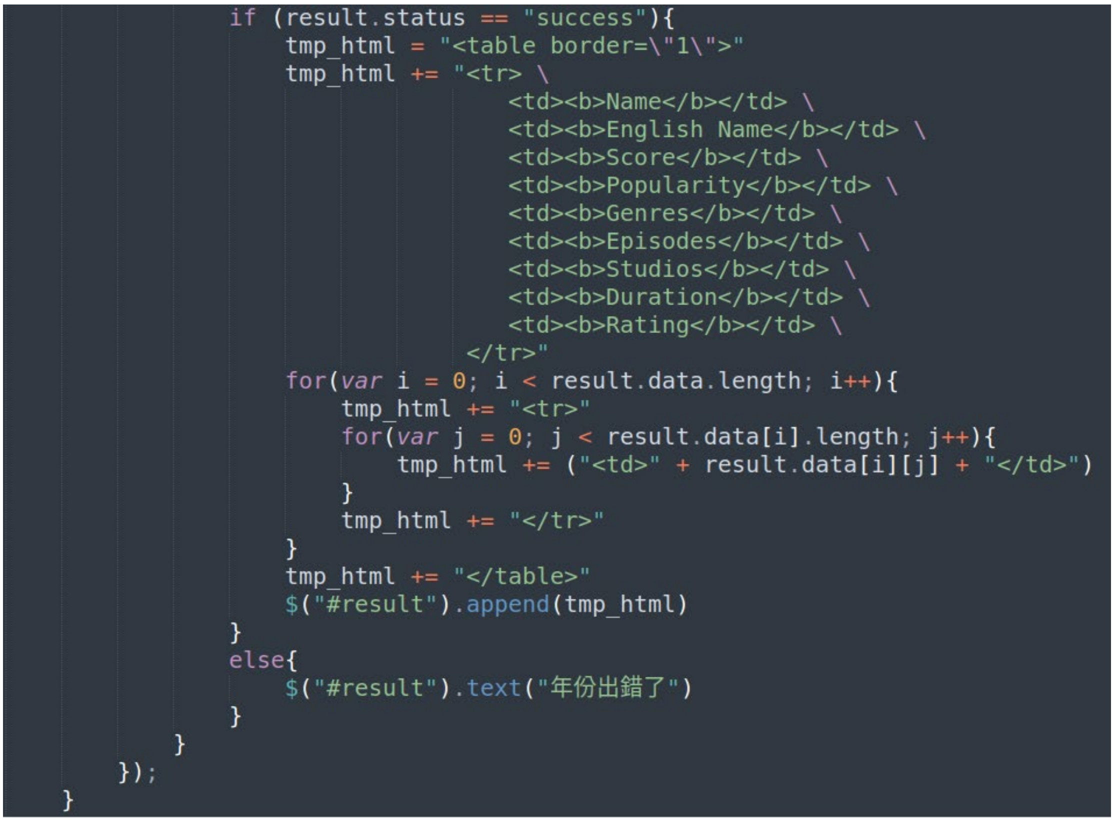
The application's functions and the related SQL queries used for the function

此 App 有兩個功能，各用一個 function 來實作。其一是列表功能，前端 function 如下 (index.html):



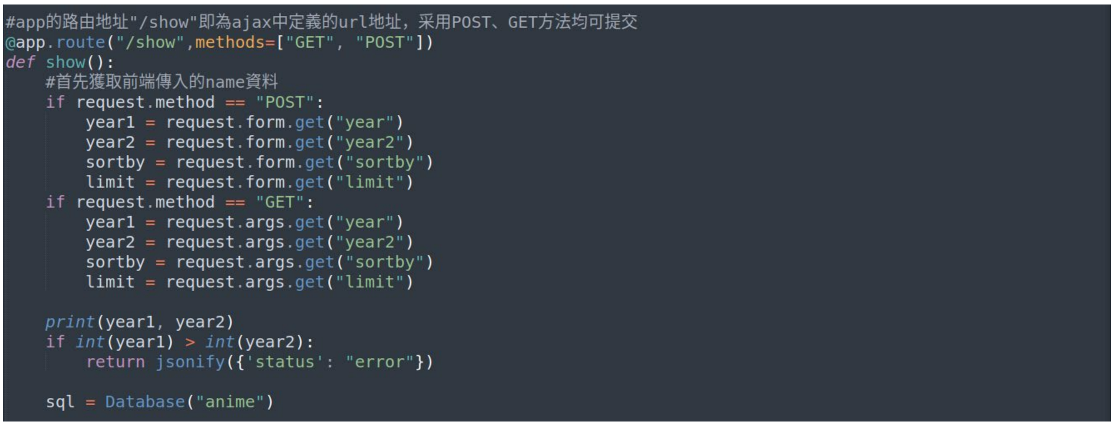
```
<script>
    function show() {
        $.ajax({
            url: "show",
            type: "POST",
            data: { "year": $("#year").val(),
                    "year2": $("#year2").val(),
                    "sortby": $("#sortby").val(),
                    "limit": $("#limit").val()
            },

            /*result為后端處理函式的回傳值！*/
            success: function (result) {
                $("#result").empty()
                //console.log(result.data)
```



```
                if (result.status == "success"){
                    tmp_html = "<table border=\"1\">"
                    tmp_html += "<tr> \
                                    <td><b>Name</b></td> \
                                    <td><b>English Name</b></td> \
                                    <td><b>Score</b></td> \
                                    <td><b>Popularity</b></td> \
                                    <td><b>Genres</b></td> \
                                    <td><b>Episodes</b></td> \
                                    <td><b>Studios</b></td> \
                                    <td><b>Duration</b></td> \
                                    <td><b>Rating</b></td> \
                                 </tr>"
                    for(var i = 0; i < result.data.length; i++){
                        tmp_html += "<tr>"
                        for(var j = 0; j < result.data[i].length; j++){
                            tmp_html += ("<td>" + result.data[i][j] + "</td>")
                        }
                        tmp_html += "</tr>"
                    }
                    tmp_html += "</table>"
                    $("#result").append(tmp_html)
                }
                else{
                    $("#result").text("年份出錯了")
                }
            }
        });
    }
```

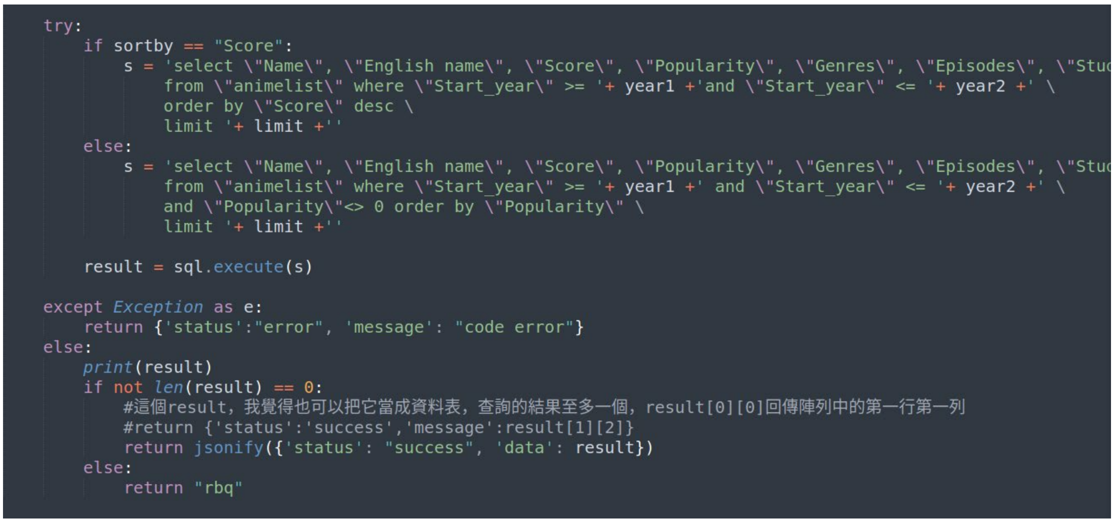
後端 flask 部分 (app.py):



```
#app的路由地址"/show"即為ajax中定義的url地址，采用POST、GET方法均可提交
@app.route("/show",methods=["GET", "POST"])
def show():
    #首先獲取前端傳入的name資料
    if request.method == "POST":
        year1 = request.form.get("year")
        year2 = request.form.get("year2")
        sortby = request.form.get("sortby")
        limit = request.form.get("limit")
    if request.method == "GET":
        year1 = request.args.get("year")
        year2 = request.args.get("year2")
        sortby = request.args.get("sortby")
        limit = request.args.get("limit")

    print(year1, year2)
    if int(year1) > int(year2):
        return jsonify({'status': "error"})

    sql = Database("anime")
```

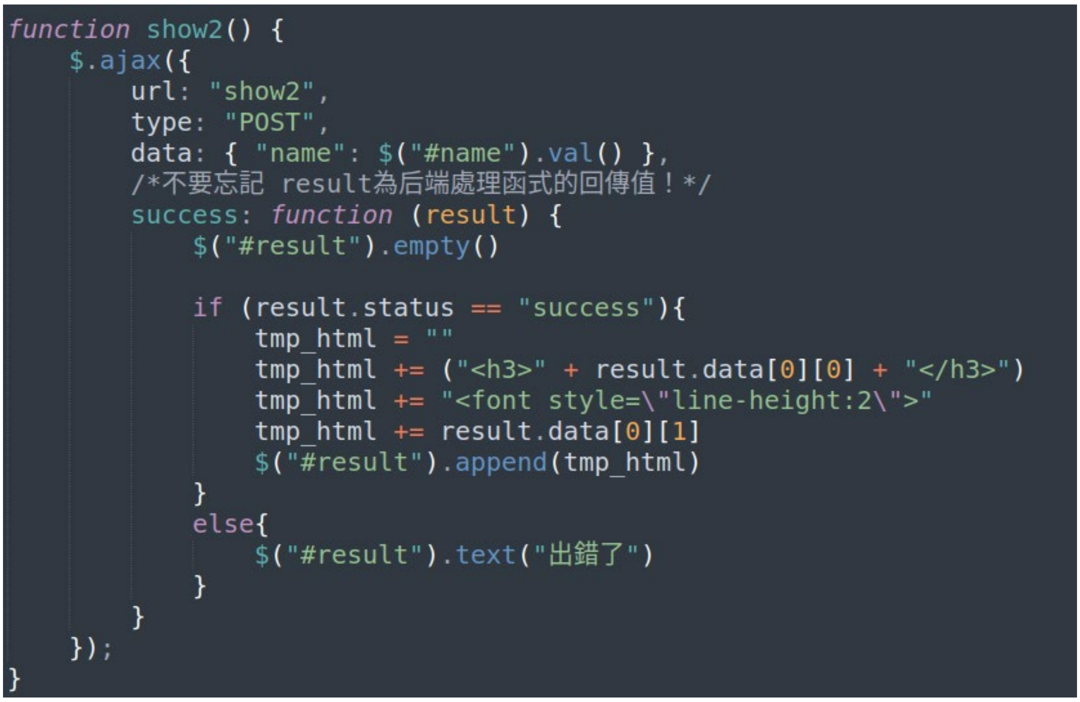


```
    try:
        if sortby == "Score":
            s = 'select \"Name\", \"English name\", \"Score\", \"Popularity\", \"Genres\", \"Episodes\", \"Stu
                from \"animelist\" where \"Start_year\" >= '+ year1 +'and \"Start_year\" <= '+ year2 +' \
                order by \"Score\" desc \
                limit '+ limit +''
        else:
            s = 'select \"Name\", \"English name\", \"Score\", \"Popularity\", \"Genres\", \"Episodes\", \"Stu
                from \"animelist\" where \"Start_year\" >= '+ year1 +' and \"Start_year\" <= '+ year2 +' \
                and \"Popularity\"<> 0 order by \"Popularity\" \
                limit '+ limit +''

        result = sql.execute(s)

    except Exception as e:
        return {'status':"error", 'message': "code error"}
    else:
        print(result)
        if not len(result) == 0:
            #這個result，我覺得也可以把它當成資料表，查詢的結果至多一個，result[0][0]回傳陣列中的第一行第一列
            #return {'status':'success','message':result[1][2]}
            return jsonify({'status': "success", 'data': result})
        else:
            return "rbq"
```

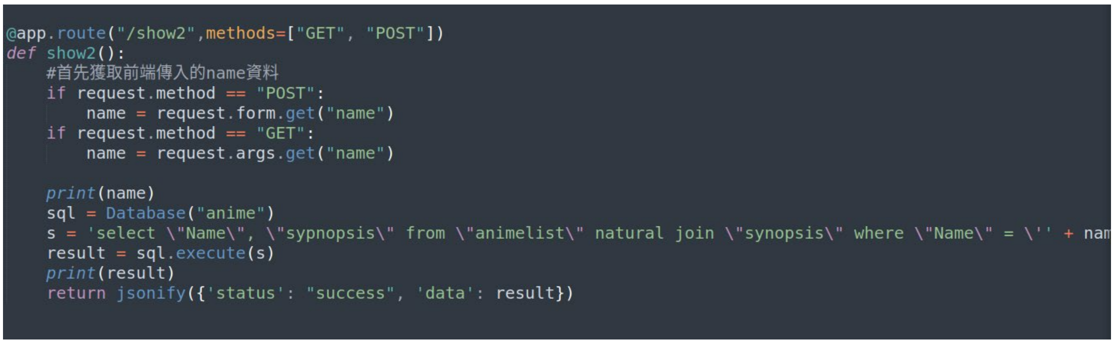
其二是搜尋摘要功能。前端 function 如下 (index.html):



```
function show2() {
    $.ajax({
        url: "show2",
        type: "POST",
        data: { "name": $("#name").val() },
        /*不要忘記 result為后端處理函式的回傳值！*/
        success: function (result) {
            $("#result").empty()

            if (result.status == "success"){
                tmp_html = ""
                tmp_html += ("<h3>" + result.data[0][0] + "</h3>")
                tmp_html += "<font style=\"line-height:2\">"
                tmp_html += result.data[0][1]
                $("#result").append(tmp_html)
            }
            else{
                $("#result").text("出錯了")
            }
        }
    });
}
```

後端 flask 部分 (app.py):



```
@app.route("/show2",methods=["GET", "POST"])
def show2():
    #首先獲取前端傳入的name資料
    if request.method == "POST":
        name = request.form.get("name")
    if request.method == "GET":
        name = request.args.get("name")

    print(name)
    sql = Database("anime")
    s = 'select \"Name\", \"sypnopsis\" from \"animelist\" natural join \"synopsis\" where \"Name\" = \'' + nam
    result = sql.execute(s)
    print(result)
    return jsonify({'status': "success", 'data': result})
```

由於在 python 語法中，字串中的引號要用反斜線 (\) 來做「逃脫」，造成可讀性降低。因此我將他們轉換成普通 PostgreSQL 語法呈現如下 (紅字表示變數):

Sort by score:

select "Name", "English name", "Score", "Popularity", "Genres", "Episodes", "Studios", "Duration", "Rating"
from "animelist" where "Start_year" >= year1 and "Start_year" <= year2
order by "Score" desc limit limit

Sort by popularity:

select "Name", "English name", "Score", "Popularity", "Genres", "Episodes", "Studios", "Duration", "Rating"
from "animelist" where "Start_year" >= year1 and "Start_year" <= year2
order by " Popularity" desc limit limit

Get synopsis:

select "Name", "sypnopsis" from "animelist" natural join "synopsis"
where "Name" = name

A demo video (less than 3 mins) with an introduction

https://drive.google.com/file/d/1ivrW43B24EZYPK6xXNlRkCwFzoczJX8V/view?usp=sharing