**資料庫管理 Final Project**

第十四組

B04902092 張均銘

B04902021 陳弘梵

**一、主題說明 - 動畫百科查詢系統**

透過自行蒐集喜歡的動畫、聲優、動畫製作、動畫歌手、動畫片頭、片尾曲資料，將其存在Mysql的資料庫內，並建造網站並提供使用者以下功能：

1. 查詢動畫的詳細資訊，包括**聲優、動畫製作、片頭片尾曲資訊。**
2. 使用者可**透過各種偏好**(如製作公司、動畫類型、編劇監督)來篩選出自己偏好的動畫列表，方便使用者查詢並整理有興趣的動畫。也可以用此**推薦使用者**他可能喜歡的動畫。

**二、開發動機**

希望能提供使用者一個更人性化的搜尋介面與搜尋方式，能夠更有條理地列出各個動畫的資訊給予使用者，讓使用者能更方便、迅速查詢到自己想要的動畫資訊，或者偏好的風格。一來因為維基百科本身頁面是沒有很美觀的，常常有太多自己其實不太需要的資訊，因為龐雜的資訊，使用者有時反而不容易找到自己想知道的資訊。二來是因為我們可以提供一個動畫搜尋跟資訊提供的整合，我們可以提供使用者依據自己喜好(風格、歌手、年份)等等資訊搜尋到符合條件的動畫並提供資訊，而維基百科並沒有辦法做到提供使用者喜好的搜尋方式，只能透過單純搜尋動畫名稱的方式。另外也希望能透過使用者的查詢分析使用者的偏好來推薦使用者可能喜歡的作品，或歌曲。

**三、使用技術**

1. 後端開發環境 : NodeJS、ExpressJS

Nodejs為一個適合前後端開發的程式語言，其程式語言裡有許多架構可進行網站的建置，如後端的ExpresJS與前端的架構ReactJS、VueJS、AngularJS，此次會選擇使用ExpressJS進行前後端的程式撰寫，並設定前端程式語法為ejs，此為ExpressJS架構下的一種前端語法，語法方面與html相差不多，唯一差別在ejs有自己的一套語法來進行前後端的資料傳遞、溝通。

1. 資料庫端環境：Mysql

透過npm package中的mysql套件來進行資料庫的安裝與架設，npm全名為nodejs package management，可以載入相當多的nodejs套件使用。相當於python的pip install。並在網站後端與資料庫相連和撰寫API來進行後端對資料庫操作，此次作業實作在windows及mac作業系統都能順利執行。

1. 資料庫端系統：Docker

Docker為應用程式虛擬化的技術，能夠處理在不同OS開發下，架設環境或webserver不相容的問題，也能減少其他使用者架設環境的困擾。舉例而言，在我們的web server下需要安裝Nodejs、mysql接著再跑sql script把data, schema匯入資料庫裡。有Docker後，使用者不用自己安裝這些套件，可以透過Docker自動把它跑起來，也不用再自行跑sql scripts匯入schema,data，也可以解決其他人架設環境不相容的問題。

1. 前端開發環境：HTML、 jQuery、 Ajax

透過後端的ExpressJS連接到前端頁面並顯示，採用了ejs格式來撰寫HTML端，並使用JavaScript的jQuery library對頁面進行操作。在與後端溝通時，則採用Ajax的技術進行post並獲取返還資料，予以呈現在HTML頁面中。我們也自行編寫配置CSS格式來美化編排頁面的呈現。

1. Github

透過Github讓不同的開發者能夠同步開發程式，並可以藉由Github的commit紀錄開發者開發的狀況與讓別人知道新加了哪些的Feature，並做好關於開發環境、使用方法的文件(README.md)

以下為github的repository:

<https://github.com/jimmychang851129/Database_Final>

**四、資料蒐集**

1. 所需蒐集屬性以及關聯:

**Animation : Studio,Episodes, Season, Rating,style(multivalued), image,AnimeDate,genre(multivalued)**

**Voice Actor : Name,Gender,Birthday, Debut date, Agent,Image**

**Agent : Name, Location, Establish date, President**

**Theme song : SongName, Open/End, Anime Name, Composer, Author, Singer**

**Singer : SingerName, Gender, Birthday, Debut date, Agent**

**Studio : Name, Location, Establish date, President**

**staff : Name, work\_company, position**

2. 屬性關聯性描述:

**動畫**有自己的名稱、為第幾季、此季有多少集數、用戶評價分數，同時也有對應的唯一製作公司，對應到多首的主題曲、多個的角色聲優、多個所屬類別。

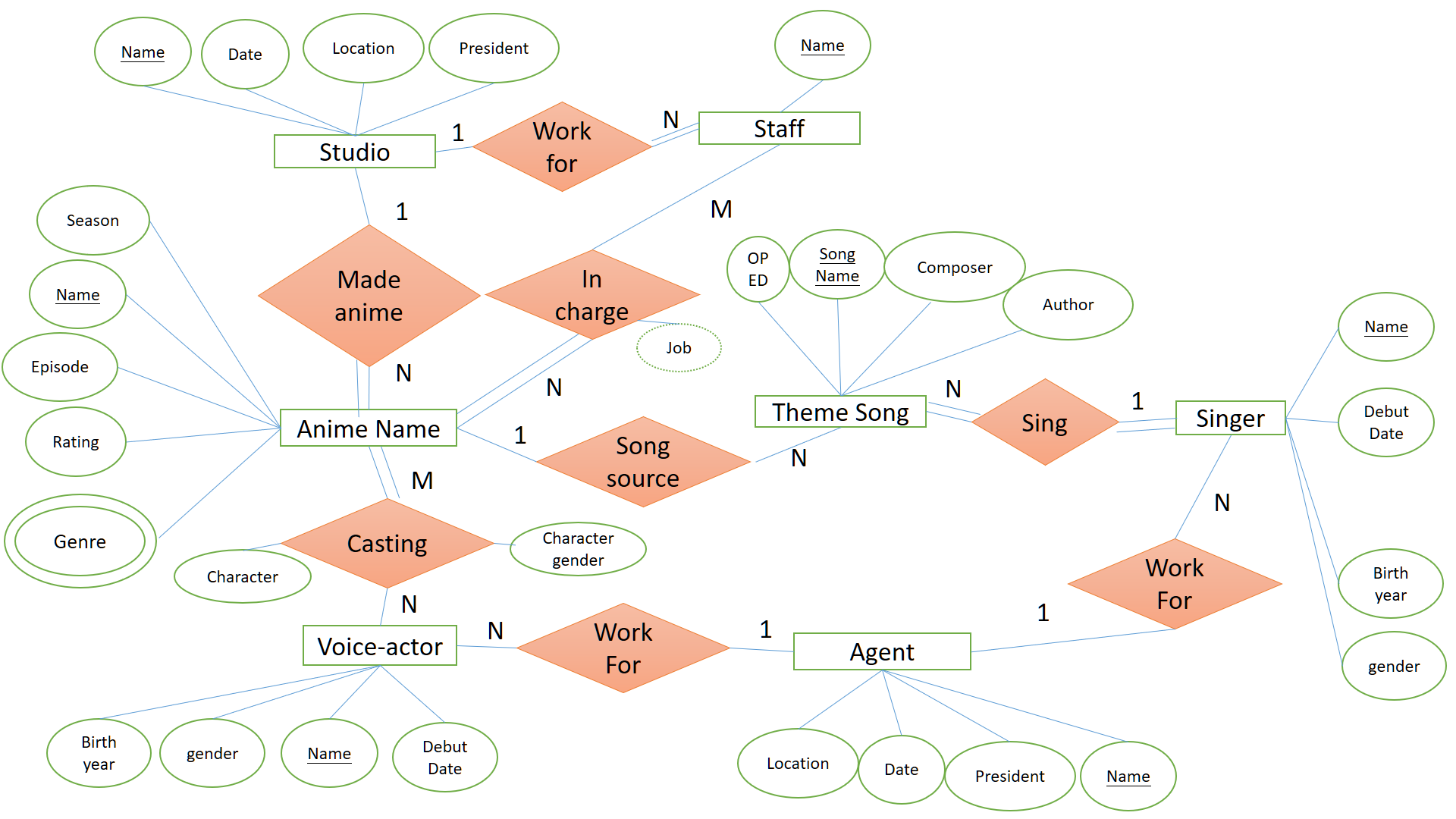
**員工**有自己的名字、隸屬於一個動畫製作公司，也負責多個動畫的多個職務。

**歌曲**有自己的歌名、作詞作曲、歌手、屬於開頭或片尾。

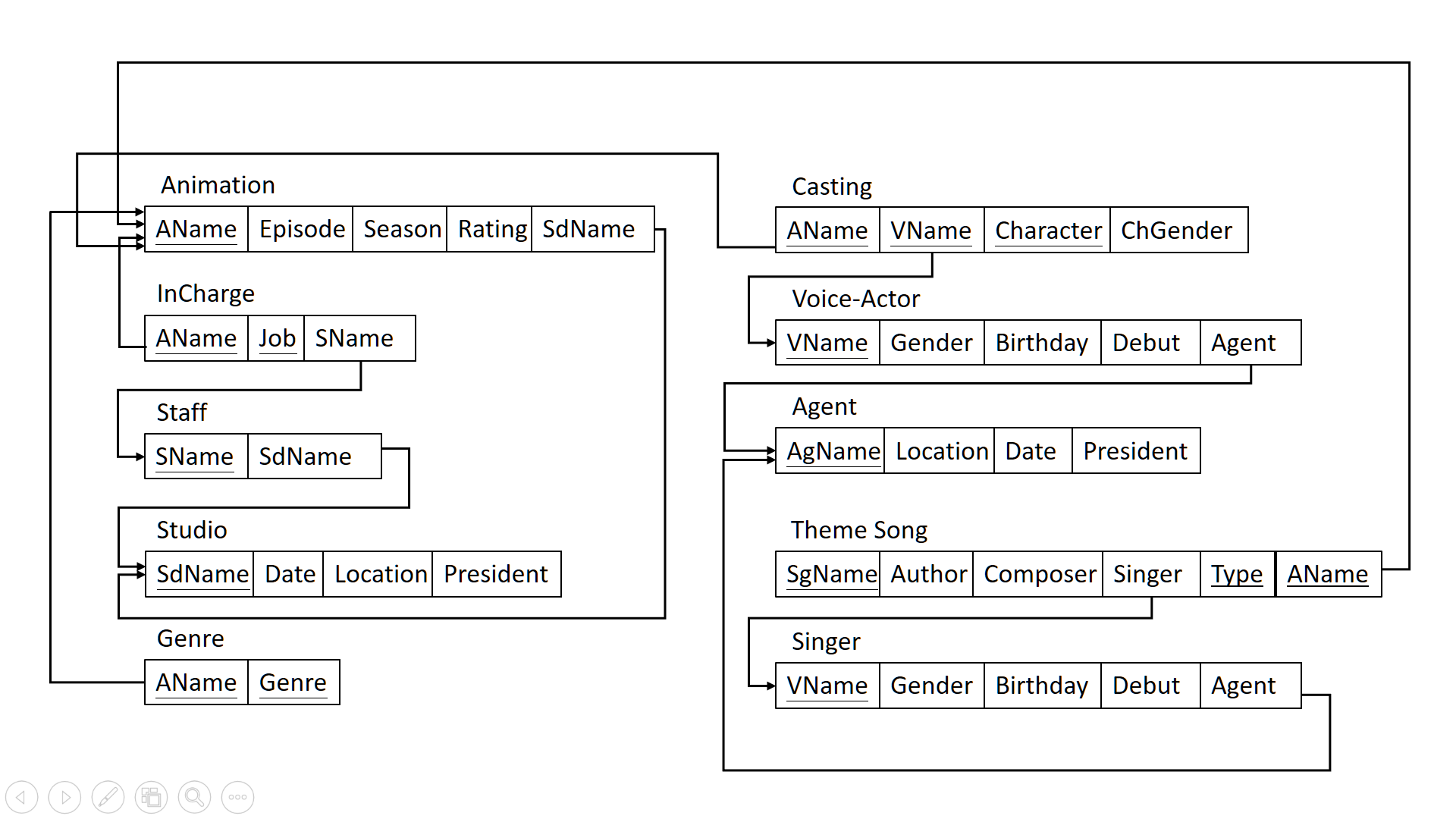
**歌手**與**聲優**有自己的名字、生日、性別、出道時間、所屬經紀公司。

**經紀公司**與**動畫製作公司**有自己的名稱、成立日期、位於地點、社長。

**五、ER diagram**

****

**六、Relational Database Schema**

****

**六、功能介紹**

參考功能連結(包含實作細節)：

<https://hackmd.io/r8ultxcQRniz1tzh15Rg0w>

主要分成四面向的搜尋

1. 動畫搜尋，顯示動畫完整資訊：
   1. 輸入動畫名稱搜尋
   2. 輸入年份搜尋
   3. 輸入風格搜尋
2. 角色搜尋，顯示角色細節與對應聲優：
   1. 輸入動畫名稱搜尋
3. 聲優搜尋，顯示聲優完整資訊：
   1. 輸入聲優名稱搜尋
   2. 輸入聲優性別搜尋
   3. 輸入出演的動畫搜尋
4. 歌曲搜尋，顯示主題曲完整資訊：
   1. 輸入歌曲名稱搜尋
   2. 輸入歌手名稱搜尋
   3. 輸入動畫的主題曲搜尋

**七、成果**

**動畫搜尋**

****

****

**角色搜尋**

****

**聲優搜尋**

****

****

**歌曲搜尋**

****

**八、Future Work**

1. **更注重資料庫安全性問題**
   1. **DOS Attack**

至於負荷量的部分因為Nodejs本身不是One Thread Per Client的Model，所以可以比原本Apache的Web server容許更多的client連線，但仍要注意DoS攻擊，做法便是多檢查API是否有哪裡的執行效率較低的部分做改善。

* 1. **SQL Injection**

Nodejs 的mysql package的function本身有抵擋掉一些簡易的sql injection，主要是可以去掉斜線的character，避免後面的指令被註解掉，但仍還有一些方法可以攻擊，因此對於輸入處可能有需要在後端多做一些處理。

1. **紀錄使用者搜尋狀況並統計**

統計使用者搜尋的狀況，分類使用者偏好哪一類型、風格的動畫並作出適當

推薦，或者喜歡的歌手、或者畫師。另外也能對動畫本身作出統計，統計搜尋該動畫的使用者男女比與其風格或歌曲、歌手之間的關係。試圖統計或找出使用者偏好的關聯性，是因為動畫風格、或者是某個畫師的風格、或者是片頭片尾曲等因素影響使用者對於動畫的興趣程度。

**九、Reference/ Data collection**

維基百科:<https://zh.wikipedia.org/wiki/%E6%97%A5%E6%9C%AC%E9%9B%BB%E8%A6%96%E5%8B%95%E7%95%AB%E5%88%97%E8%A1%A8>

巴哈姆特動畫瘋:<https://ani.gamer.com.tw/index.php>

Express and mysql : <https://dotblogs.com.tw/explooosion/2016/07/18/010601>

mysql docker設定 : <https://medium.com/@lvthillo/customize-your-mysql-database-in-docker-723ffd59d8fb>

mysql 指令 : <http://www.mysqltutorial.org/mysql-show-columns/>

mysql utf8設定 : <https://dev.mysql.com/doc/refman/5.7/en/charset-server.html>

jQuery 指令：<https://jquery.com/>