**資料庫管理 Final Project Proposal**

第十四組

B04902092 張均銘

B04902021 陳弘梵

B04902020 張耀中

**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**

1. 建立一個動畫百科查詢系統，提供一個整合所有資訊的平台供使用者查詢動畫的所有相關資訊與分類。
2. 提供使用者依據自己喜歡的類型(熱血、搞笑、戀愛)來**分類動畫**，未來可透過演算法藉由使用者選擇的種類分析使用者喜歡的類型並作推薦。
3. 提供一個平台使使用者能夠了解當今動畫產業的趨勢變化，包括每年的動畫製作的多寡、各家作畫公司不同年度的產出或每年大眾對於動畫類型的偏好(透過當年動畫產出多為哪種類型判斷)。
4. 透過資料的整合，提供其他包含歌曲、聲優等等元素與動漫產業鏈上的合作情形與趨勢。

**三、使用技術**

1. 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. Mysql npm package

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

**四、Data Requirements**

1. possible Relations & Attributes:

**Anime Name : Studio(動畫製作),Episodes, Season, Rating,style(multivalued), Theme song, Genre, Voice Actor**

**Voice Actor : Birthday, Debut date, Agent(經紀公司)**

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

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

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

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

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

2. Attribute dependencies Description:

**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、多個的腳色聲優。

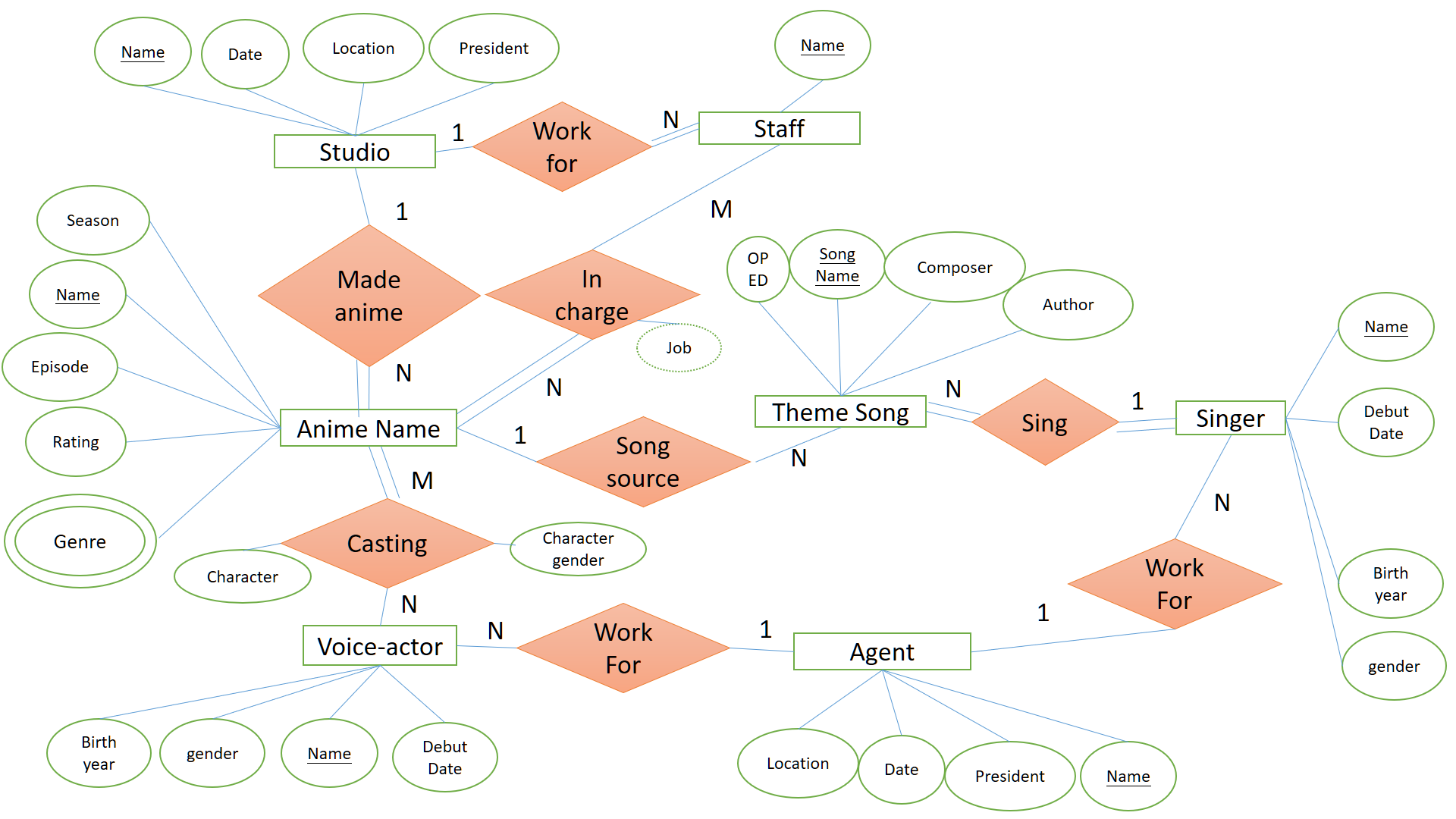
**員工**有自己的名字、隸屬於一個動畫製作公司，也負責多個動畫的多個職務。

**歌曲**有自己的歌名、作詞作曲、歌手、屬於開頭或片尾。

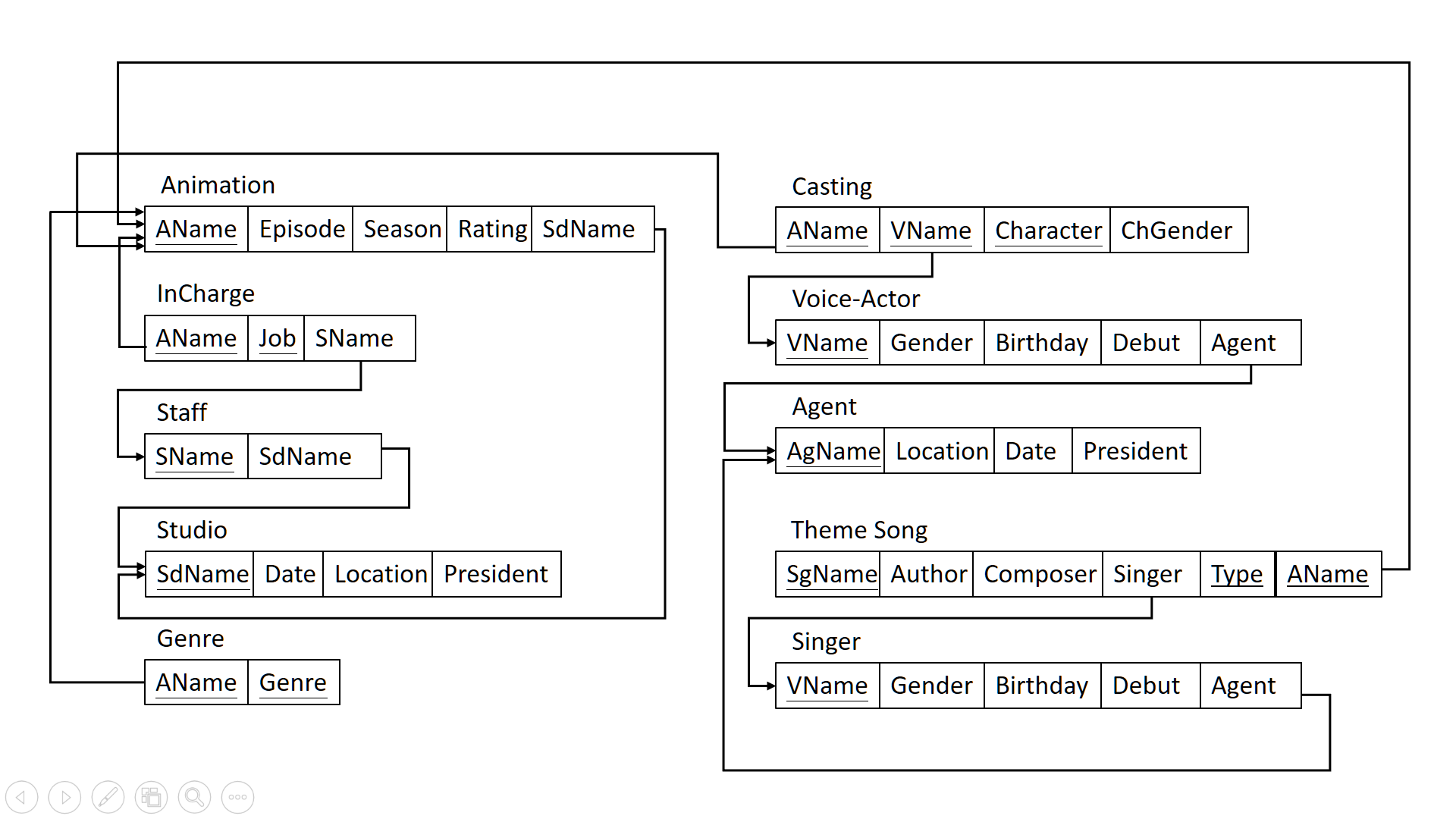
**歌手**與**聲優**有自己的名字、生日、性別、出道時間、所屬經紀公司。

**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**

****

**七、Functional Requirements and Pseudo Code**

**查動畫名**

input : 動畫名

output : 所有動畫資訊

psuedo code:

select \* from anime where anime==inputname

**查動畫類型**

input : 動畫類型(療癒、校園、運動)

output : 動畫類型為該類型的動畫，count of anime

psuedo code:  
 1.join anime name和Genre兩個relation

2.select 符合input類型的動畫，並輸出動畫的資訊

3. 用COUNT函式計算總共有幾部動畫符合這個限制

**查季數、集數**

input : 動畫總季數/總集數

output : 季數/集數符合此數量的動畫的詳細資訊

psuedo code:

1. select \* from anime where 集數和input一樣

**查聲優配過的動畫**

input : 聲優名

output : 所有這聲優有配過的動畫資訊，count of anime

psuedo code:

1. 有一個relation專存聲優與動畫的特徵(因聲優與動畫關係為M:N，因此relation schema裡中間有一個table)
2. 搜尋到關係後將動畫詳細資訊輸出，和配過的動畫的數量

**查動畫公司作品**

input : 動畫公司名稱

output : 所有該動畫公司作品及資訊

psuedo code:

1. join動畫名和公司這兩個relation
2. select 動畫公司 where 動畫公司名 = input動畫公司名

**查季度動畫，與該季度某類型動畫**

input : 年月份，類型(optional)

output : 該年、該月份符合類型的所有動畫

psuedo code:

1. join anime name和genre的relation
2. 如果類型為NULL直接select date符合的資料
3. 如果類型非NULL，搜尋date和類型都符合的資料

**查動畫所有聲優**

input : 動畫名

output : 該動畫內所有的聲優成員

psuedo code:

1. join Anime\_name和Voice\_actor的relation
2. select Voice\_actor name where Anime\_name name = input name

**查動畫所有聲優**

input : 經紀公司名

output : 隸屬於該經紀公司的所有聲優、歌手

psuedo code:

1. join Voice actor 和 Agent的relation為A
2. join Singer 和Agent的relation為B
3. select Voice actor Union Singer where input名=經紀公司名

**某歌手唱過的動畫歌**

input : 歌手名

output : 該歌手唱過的所有歌、歌的來源動畫、為op或ed

psuedo code:

1. join Anime\_name與Voice-actor間延伸出來的relation和Anime\_name
2. 查詢singer = input的歌手名，並將所需的attribute萃取出來

**查某動畫的動畫歌**

input : 動畫名稱

output : 歌手資訊、曲名

psuedo code:

1. join Anime\_name和Theme\_song成relation A
2. join relation A與 Singer
3. 查詢動畫名稱=input動畫名稱，將需要的歌手、歌曲資訊取出

**八、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>