

# DBMS Final Project Document

## 一、系統說明

### 1. 架構與環境

- 使用者介面：Windows Form .net
- 資料庫：MySQL
- 使用 C# 將內嵌或是使用者輸入的 SQL 指令傳送到 MySQL

### 2. 介面說明

- 左圖為起始畫面，點選選單可選擇功能(如右圖)

音樂播放軟體資料庫

SQL 指令:

傳送

查詢結果

音樂播放軟體資料庫

- MYSQL
- SELECT-FROM-WHERE 專輯中的歌曲與年份
- DELETE 刪除使用者喜歡的歌
- INSERT 新增使用者喜歡的歌
- UPDATE 使用者更改名字
- IN 播放清單共編者名字和ID**
- NOT IN 別錯過這些台語歌
- EXISTS 相關的播放清單
- NOT EXISTS 沒有喜歡日文歌的使用者
- COUNT 中文歌排行榜(依播放次數)
- SUM 歌手專輯總長度(秒)
- MAX 歌手最新發行
- MIN 歌手最新專輯
- AVG 使用者六月平均每日播放歌曲數
- HAVING 使用者年度喜愛歌手(聽歌5次以上)

查詢結果

- 可選擇使用的指令並按下傳送，底下的 label 就會顯示該按鈕預設的指令，查詢結果以下方表格呈現(如左圖)；也可以選擇 MYSQL 自行輸入指令，下方也會產生輸入的指令與結果(如右圖)。

Form1

音樂播放軟體資料庫

MAX 歌手最新發行

傳送

SQL 指令:

SELECT albumYear, albumName, artistName, albumType  
FROM album, artist, (SELECT MAX(albumYear) m FROM album  
WHERE albumArtistId = '0' GROUP BY albumArtistId)M  
WHERE M.m = albumYear AND albumArtistId = '0' AND  
albumArtistId = artistId;

查詢結果

	albumYear	albumName	artistName	albumType
▶	2020	溫柔 #MeydayEl...	五月天	單曲
*				

Form1

音樂播放軟體資料庫

MYSQL

SELECT albumYear, albumName, artistName, albumType  
FROM album, artist, (SELECT MAX(albumYear) m FROM album  
WHERE albumArtistId = '1' GROUP BY  
albumArtistId)M  
WHERE M.m = albumYear AND albumArtistId = '1' AND  
albumArtistId = artistId;

傳送

SQL 指令:

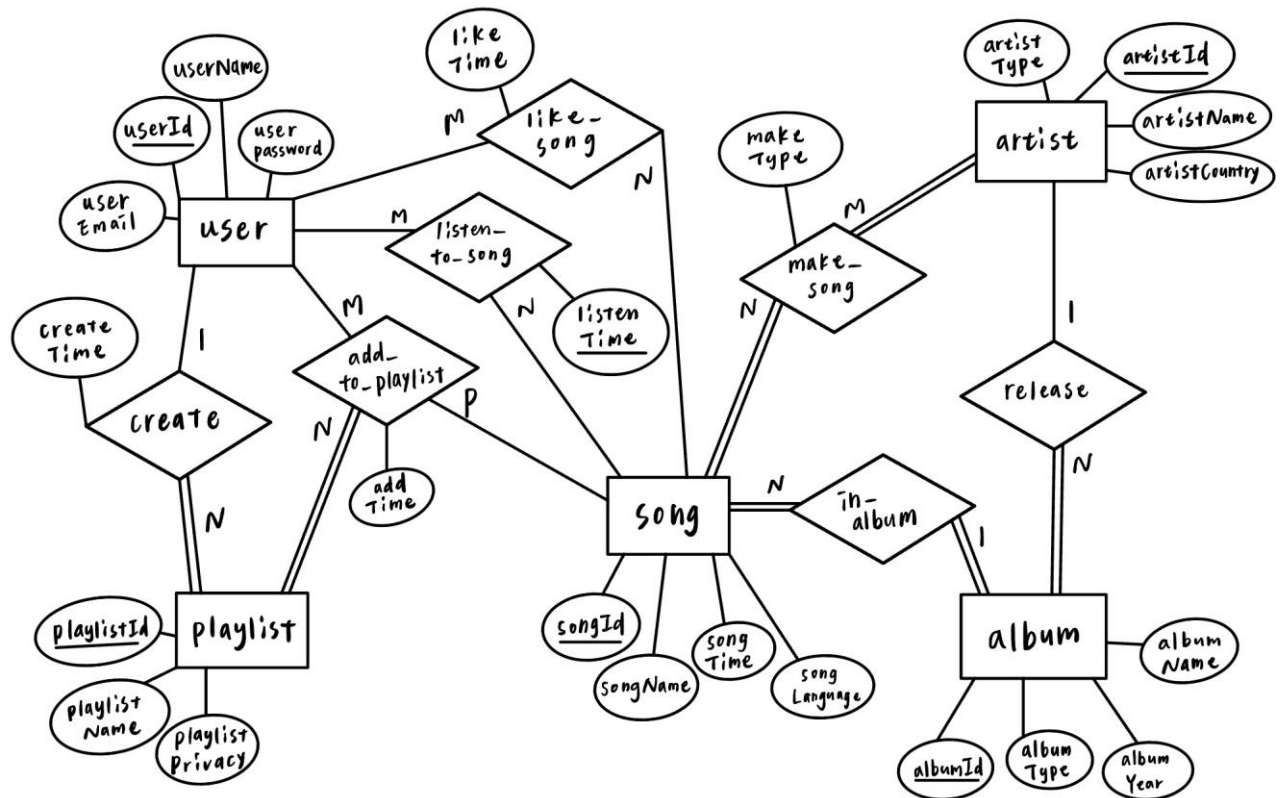
SELECT albumYear, albumName, artistName, albumType  
FROM album, artist, (SELECT MAX(albumYear) m FROM album  
WHERE albumArtistId = '1' GROUP BY albumArtistId)M  
WHERE M.m = albumYear AND albumArtistId = '1' AND  
albumArtistId = artistId;

查詢結果

	albumYear	albumName	artistName	albumType
▶	2021	創作	ヨルシカ	迷你專輯
*				

## 二、資料庫設計

### 1. ER diagram 與 relation schema



user

<u>userId</u>	userName	userEmail	userPassword
---------------	----------	-----------	--------------

playlist

<u>playlistId</u>	playlistName	playlistPrivacy	creatorId	createTime
-------------------	--------------	-----------------	-----------	------------

song

<u>songId</u>	songName	songTime	songLanguage	songAlbumId
---------------	----------	----------	--------------	-------------

album

<u>albumId</u>	albumName	albumType	albumYear	albumArtistId
----------------	-----------	-----------	-----------	---------------

artist

<u>artistId</u>	artistName	artistCountry	artistType
-----------------	------------	---------------	------------

add-to-playlist

<u>userId</u>	<u>playlistId</u>	<u>songId</u>	addTime
---------------	-------------------	---------------	---------

like-song

<u>userId</u>	<u>songId</u>	likeTime
---------------	---------------	----------

listen-to-song

<u>userId</u>	<u>songId</u>	listenTime
---------------	---------------	------------

make-song

<u>artistId</u>	<u>songId</u>	makeType
-----------------	---------------	----------

### 2. 說明

#### ■ Table: user

- userId → 使用者 ID(key)
- userName → 使用者名字
- userEmail → 使用者信箱
- userPassword → 使用者密碼

#### ■ Table: playlist

- playlistId → 播放清單 ID(key)
- playlistName → 播放清單名字
- playlistPrivacy → 播放清單隱私(分為私人或公開)
- userPassword → 播放清單建立者 ID，因為 user 和 playlist 是 1:N 的關係，因此在這邊存放建立者 ID 與建立時間
- createTime → 播放清單建立時間

#### ■ Table: song

- songId → 歌曲 ID(key)
- songName → 歌曲名字
- songTime → 歌曲長度(為了方便計算以秒為單位)
- songLanguage → 歌曲語言

- songAlbumId → 歌曲存放的專輯 ID，因為 album 和 song 是 1:N 的關係，因此在這邊存放 album 的 ID

#### ■ Table: album

- albumID → 專輯 ID(key)
- albumName → 專輯名字
- albumType → 專輯類型(有專輯、迷你專輯、單曲等)
- albumYear → 專輯發行年分
- albumArtistId → 發行專輯的藝人 ID，因為 artist 和 album 是 1:N 的關係，因此在這邊存放 artist 的 ID

#### ■ Table: artist

- artistID → 藝人 ID(key)
- artistName → 藝人名字
- artistCountry → 藝人國家
- artistType → 藝人類型(有個人、團體等)

#### ■ Table: add\_to\_playlist

某使用者新增某首歌到某播放清單。因為播放清單可以共同編輯，所以

user、playlist、song 是三元關係 M:N:P，因此獨立拉出一個 table 存放。

- userID → 使用者 ID(key)
- playlistId → 播放清單 ID(key)
- songId → 歌曲 ID(key)
- addTime → 新增至播放清單的時間

#### ■ Table: like\_song

某使用者喜歡某首歌。因為 user 喜歡 song 的關係是 M:N，所以獨立拉出一個 table 存放。

- userID → 使用者 ID(key)
- songId → 歌曲 ID(key)
- likeTime → 按下喜歡的時間點

#### ■ Table: listen\_to\_song

某使用者聆聽某首歌。因為 user 聆聽 song 的關係是 M:N，所以獨立拉出一個 table 存放。

- userID → 使用者 ID(key)
- songId → 歌曲 ID(key)

- listenTime → 按下播放的時間點(Key)，因為同一位使用者可能聆聽同首歌好幾次，需要分開計算，因此聽的時間也一起拿來當 key

#### ■ Table: make\_song

某藝人製作某首歌。因為一首歌可能有多個歌手合唱，所以 artist 製作 song 的關係是 M:N，因此獨立拉出一個 table 存放。

- artistID → 藝人 ID(key)
- songId → 歌曲 ID(key)
- makeType → 製作的型態(有獨唱、合唱等)