1. Project Descriptions:

(1) The background of your database application.

建立一個可以查詢電影相關資料的網站，可由國家、導演、工作室、獎項、演員、評分……去搜尋出各個電影的資訊，除此之外還能夠進行新增及刪除，並透過圖形化的介面讓使用者更方便管理系統。

我們還有製作登入系統，只有root能夠新增與刪除資料，一般使用者則無

(2) Elaborate on 5 useful queries for your database application.

以下是我們舊的5個query:

1.查詢某個國家所拍攝的電影

2.查詢特定評分的電影

3.查詢特定時長的電影

4.查詢某個導演特定評分的電影

5.查詢特定年份中評分大於8的電影

**由於系統更加完善，我們將展示新的5個更複雜的query:**

**1.查詢某個工作室所拍攝，某個國家所發行，某年後上映，在特定時長，並且在特定評分之間的電影**

**2.查詢相似名稱，特定預算範圍，且特定類型的電影**

**3.查詢某個導演所拍攝，特定類型，特定年齡限制，並獲得特定獎項的電影**

**4.查詢所屬某個工作室，並執導過某部(或相似)電影的導演，且導演性別必須是男性**

**5.查詢有演過某部(或相似)電影，特定國籍，且相似名字的女演員**

我們最後做出來的結果上述5個新queries都可以被查找出來，除此之外我們把我們的查詢系統分成7個部分，分別是

1.電影

2.演員

3.導演

4.工作室

5.評論:

6.Casting

7.獎項:

除了上述的5個queries，我們也可以從其他7個部分去得到更多詳細的資訊，分成不同的主體去做查詢，也方便使用者對資料進行刪除，而每個查詢系統實際的輸入值，我們會在後面的功能介紹，詳細說明

我們從原本設計的queries為基礎，創建了更多可以被查找的資訊，並從這些資料做延伸，做出屬於他們的查詢系統，而系統裡面會有更多細部的資料ex:導演 演員，工作室等查詢系統，除了查詢資料外，我們還有新增資料和刪除資料的功能，可以藉由這兩個功能，擴大或縮小資料庫

5個新queries的個別查詢結果展示:

1. 查詢某個工作室所拍攝，某個國家所發行，某年後上映，在特定時長，並且在特定評分之間的電影

例如我想尋找Marvel工作室所拍攝，USA所發行，在2010年後上映，不超過150分鐘，平均評分在5~8之間的電影



1. 查詢相似名稱，特定預算範圍，且特定類型的電影

例如我想尋找有關於蜘蛛人的電影，所以電影名稱那邊打上spider，而類型方面我限定是Action，最後用預算的範圍來做檢索



1. 查詢某個導演所拍攝，特定類型，特定年齡限制，並獲得特定獎項的電影

例如我想尋找John Moore所執導的電影，而且要是動作片，而且需要17歲以上才能觀看，並且需要獲得Good Sound這個獎項的電影



1. 查詢所屬某個工作室，並執導過某部電影的導演，且導演性別必須是男性

以下的例子中，我想查詢有哪個導演是Marvel工作室的，而且性別是男性，而且還有執導過蜘蛛人的相關電影(輸入spider去找關聯)，最終會輸出有哪些導演，並且提供其性別、工作室與所以執導過的電影



1. 查詢有演過某部(相似)電影，特定國籍，且相似名字的女演員

例如我想尋找有演過Stars Wars系列的女演員，且國籍必須是USA，並且名字必須和我所檢索的Daisy相似，最終會輸出來有那些演員，其性別與國籍，還有參演過哪些電影。

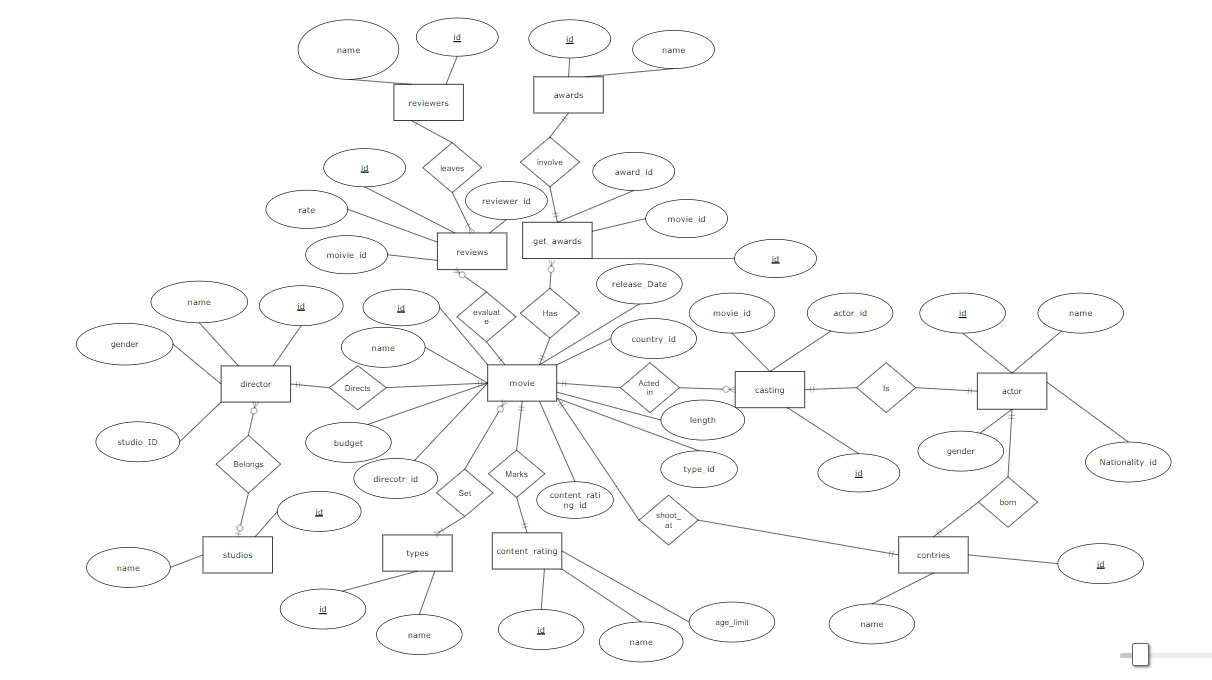


(3) Explain the design of your ER diagram for the database application.

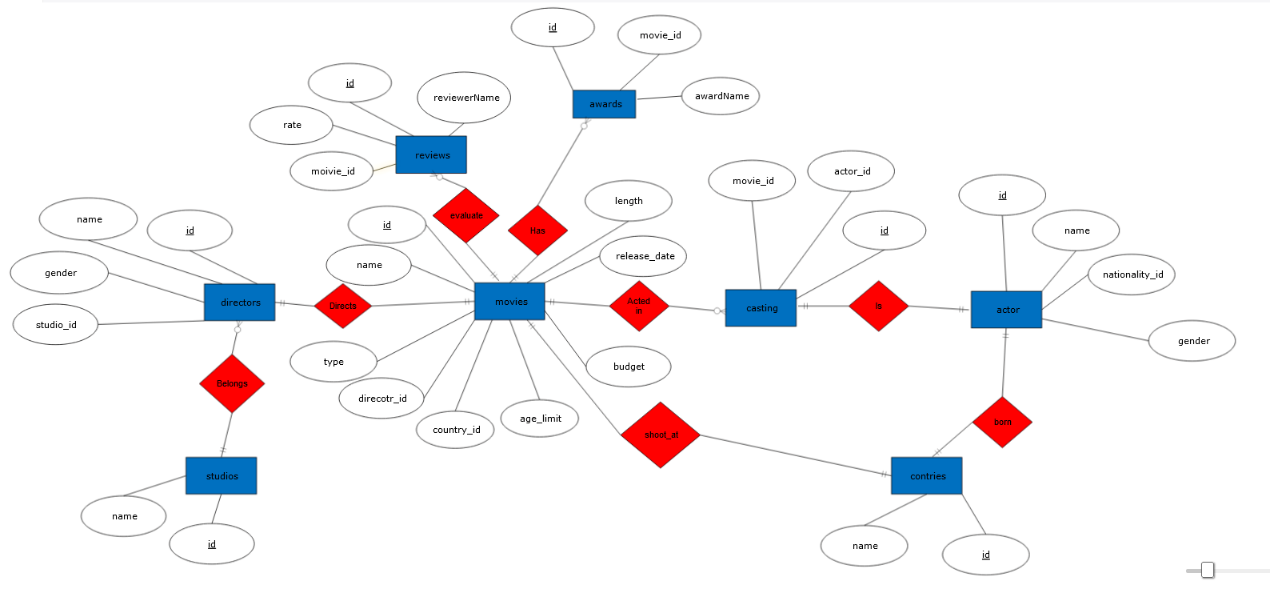
我們將movies作為核心的資料，先分支出castings，代表一個電影會有0個或多個選角，每個選角再去對應到actors，取得每個演員的資訊。我們將種類這個資訊結合到movies這個table中，電影有可能是動作片、喜劇、愛情片……，每部電影會對應到一個種類，在新增電影時，可從三十幾個種類中去選擇。我們最後也將觀看的年齡限制結合到movies這個table中，在新增電影時有0,6,13,17這四個不同的選擇，代表需要至少幾歲才能夠觀看此電影。directors是電影的導演，每部電影都會有一個主導演，但一個導演可主導多部電影，directors會再與studios做連接，看是隸屬於哪個工作室。awards為電影所獲得的獎項，有可能電影並沒有得獎。而reviews是觀看者為電影所留下的評分，檢索電影所出現的評分會從裡面去做搜尋，並算出平均分數來顯示。

(4) E-R Diagram: You can use any tool to draw the diagram, but be careful to use proper shapes and links according to the applied data model (e.g., Chen, IDE1FX, UML).

(此為舊資料)

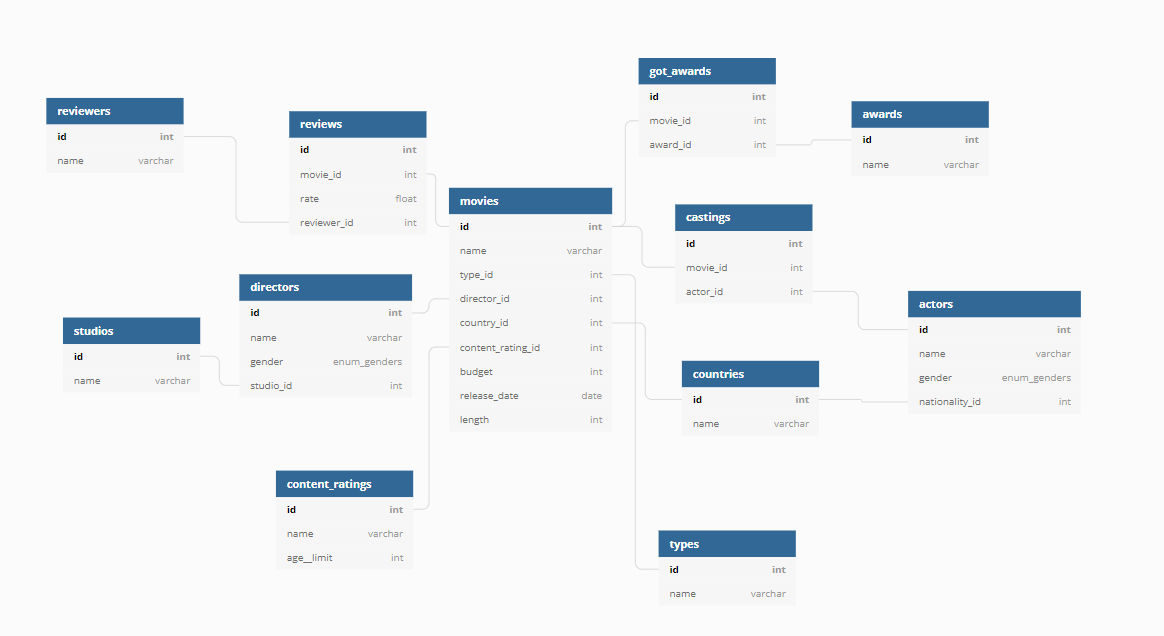


(此為新資料)

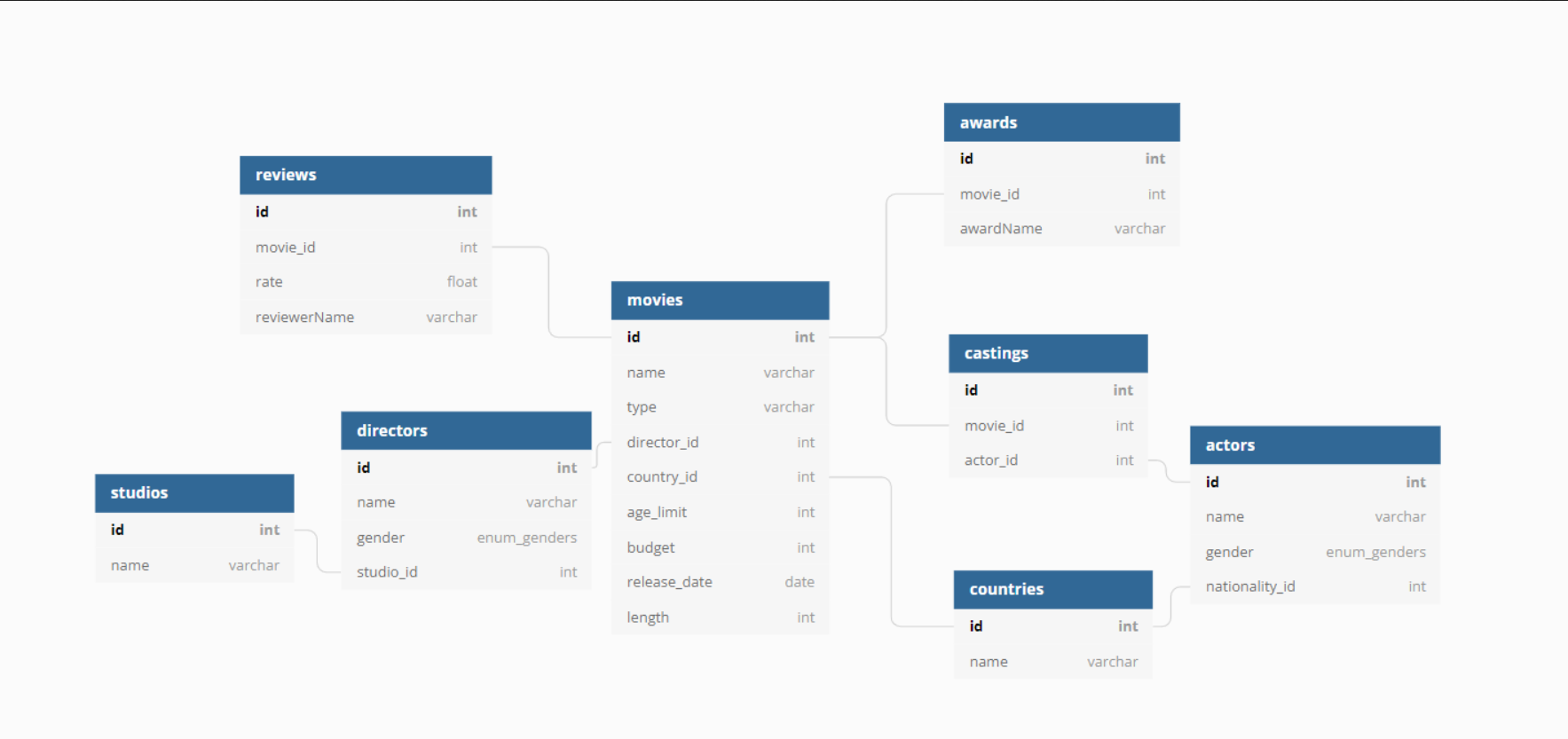


3. Database Schema: You should then convert the E-R diagram into several schemas including table name and attributes. The primary key should be properly specified in each schema. Please also provide some descriptions to illustrate the definition for each of your schemas. For example, why the attribute set you choose can be a primary key?

(此為舊版Relation Schema)



(修正後最後實際使用的Relation Schema)



資料庫正規化的目的，是為了減少重複性資料頻繁出現，使資料能井然有序地儲存，同時提升資料庫操作效率、減少資料在更新時出現異常，考慮到movies核心資料表中的原始資料有過多重複的屬性，因此共分割為8個資料表。為使資料表間的屬性方便對應，雖對於部分資料表而言，可使用複合屬性作為單一Primary Key使用，但考慮欲使用數字對應兩資料表間的某欄位，每個資料表均使用auto increase id屬性作為Primary Key。

• 新版中將reviewers table跟 reviews table結和成reviews table，由於我們只在乎評分，是哪個人留言並不是重點，因此捨棄另外維護一個reviewers，這麼做也讓實際撰寫sql時較容易。

• 新版中將電影分級、電影類型塞到movies裡，因為這些資訊會提供下拉式選單，不需擔心insert錯誤的情況，多一個table只會讓檢索更為複雜。

• 新版中將got\_awards table跟award table 合成 award table，也是一樣不需再額外分出一個table，用一個table反而檢索效率較高。

**下方紅色的字，代表和原本設計的table差異的部分**

movies:電影的核心資訊，包括名稱、預算、上映日期、片長、類型、節目分級(分別為type和content\_rating合併來的)。主要導演id、出品國家id均Foreign Key，對應到其對應的資料表的id屬性。在這個資料表中，可以有很多同類型的電影、同個導演拍攝的作品、同個國家出品的作品、同個節目分級級別的電影。

directors:導演的基本資訊，包括姓名、列舉的性別、所屬工作室ID，且為Foreign Key，對應到studios資料表的id屬性，且1個導演只會隸屬於1間工作室，檢索時可透過導演來查出某個導演有執導哪些電影，或是透過電影去尋找導演。

studios:導演所屬的工作室的相關資訊，包括工作室名稱，1個工作室可能會有很多位隸屬的導演，故id屬性為directors資料表中的director\_id屬性所對應。

content\_ratings (與movies合併，age\_limit塞到movies)

將節目分級表刪除塞到movies裡，可以減少冗余資料的儲存與搜索，不用多存id跟name，檢索效率較高，可透過movies table中的age\_limit直接取得年齡限制。

reviews:評論者對於某部電影的評分資料，主要為評分數據，而movie\_id對應到movies資料表的id屬性，為Foreign Key。後面不考慮到底是哪個評論者去留言，只要知道留言者的名字以及他給的分數就可以，簡化我們實際撰寫sql的複雜度，也提升檢索的效率。

reviewers (合併到reviews)

reviews 原本存 reviewers\_id，但是這麼分太細了，要找評論者的名字還要到reviewers搜尋，不但費時間而且存了多餘的資訊，故合併到reviews中，該筆review會直接顯示評論者的名字。

types (合併到movies)

將電影的種類直接存到movies table中，可以減少多餘的資料，我們直接提供30幾種類型(下拉式選單)供使用者查詢，查詢時只需查看movies table即可，降低撰寫sql的複雜度。

get\_awards(合併到awards)

跟reviewers相似的狀況，直接將獎項的名字合在一起即可，故合併到awards中

castings: Pivot-Table，movie\_id屬性為movies資料表中的id屬性的Foreign Key、actor\_id屬性為actors資料表中的id屬性的Foreign Key，表示一部電影出演的一位演員，故設置UNIQUE INDEX(movie\_id, actor\_id)，限制同部電影只能有同樣的一位演員。每部電影得以紀錄超過一位以上的出演演員。

awards:獎項的基本資訊，包括獎項名稱和movie\_id(get\_awards合併來的)。

actors:演員的基本資訊，包括姓名、列舉的性別、國籍ID，其中國籍ID為Foreign Key，對應到countries資料表的id屬性，且1個演員只會隸屬於1個國籍。

countries:國家相關的基本資訊，包括國家名稱，其id屬性是movies資料表中country\_id屬性的Foreign Key，表示電影出品國家，也是actors資料表中nationality\_id屬性的Foreign Key，表示演員國籍。由於國家不會衝突名稱，故設置UNIQUE INDEX(name)。

4. Non-trivial Functional Dependencies: Analyze each schema of your database and provide its non-trivial dependencies. For each schema, specify which normal form does it holds.

每個資料表中，每個屬性都是不可分割的原子屬性，故必定符合1NF，又每個資料表的Primary Key都只有1個欄位，則必定符合2NF，且不會符合BCNF。故若非主屬性不具備對候選鍵的傳遞依賴，則該資料表符合3NF，反之符合2NF。

綜上，屬於2NF的資料表有: movies, actors, directors屬於3NF的資料表有: awards, castings, countries, reviews, studios。

5.Find資料的方式:

我們查找資料的方式，是從imdb電影查詢網站裡面的排行榜前50大電影為基礎，並從每一部電影的導演、演員、工作室，去延伸出更多的電影，演員，工作室，導演……這樣處理的話可以增加每個資料的相關性，再進行資料檢索的時候，就可以透過這之間的關聯搜索到不同的結果，如此一來就不會出現都是亂數產生而資料之間沒有關聯的結果。

Imdb參考網址: <https://www.imdb.com/chart/top/>

Movies , studios , directors , castings , country , type皆是從imdb電影介紹裡面爬蟲找尋出來的

然而awards ,reviews則是使用程式碼 跑亂數產生出來的

由於我們必須讓資料之間有關聯，所以我們會手動調整資料，例如castings需要的是演員的id和電影的id，因此我們必須比對兩者的id後再進行資料的建構，這樣id才會對應正確，其他資料亦是如此，在建構時我們也會盡量讓演員有演出不同的電影，導演有執導不同的電影，工作室也有不同的導演等等，讓關聯性增加

以刺激1995為例:

我們可以藉由爬details , Box office , Technical spaces得到各個table需要的資料，並把她個別分配寫入資料所屬的table

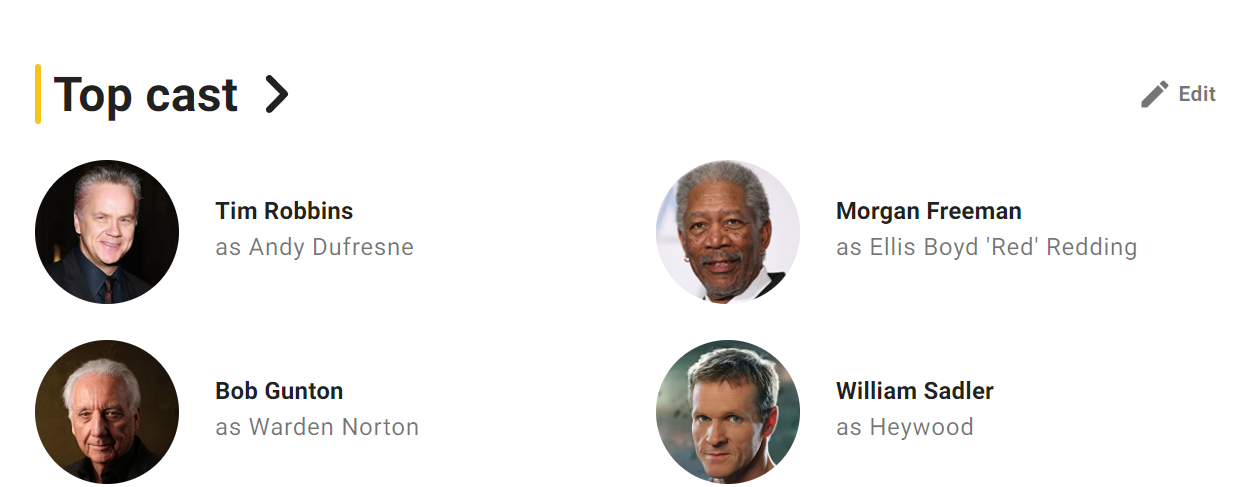




而Casting的資料則是從top cast爬來的，其中我們每一部電影大約就取最前面

的兩位演員，並從演員資料，在做一次爬蟲，把演員演過的熱門電影當作資源

寫入table，之後再手動調整id



刺激1995參考網址: <https://www.imdb.com/title/tt0111161/?pf_rd_m=A2FGELUUNOQJNL&pf_rd_p=9703a62d-b88a-4e30-ae12-90fcafafa3fc&pf_rd_r=YTN1C61CMJE1KD5AEGBM&pf_rd_s=center-1&pf_rd_t=15506&pf_rd_i=top&ref_=chttp_tt_1>

功能介紹

主要功能:藉由使用者輸入想要找尋的相關資訊，查找出相對應的資料，一

共有分7個主體可以提供搜索，搜索完成後也可以點旁邊的按鈕進行刪除

若有其他table對這項刪除的資料有foreign key的關係，也會一同刪除。

接下來我們會有更進一步如何使用我們系統的步驟

1. **電影:**

可根據圖中共12個參數進行調整，搜索後會給予符合所有條件的表格。

(如果沒有輸入的話則會使用預設值)



若該輸入數字的地方亂打的話則會跳出錯誤訊息，有對檢索的資料做防呆



以下的結果是搜尋出電影名稱中有the的電影



當我把第一個The Adventure of Robin Hood刪除再進行一次檢索後可發現找不到了，而這個電影相關的castings，reviews也會一同刪除



再來我們可以新增電影的資料



這邊也有做防呆，如果有資料沒有輸入的話會要求輸入



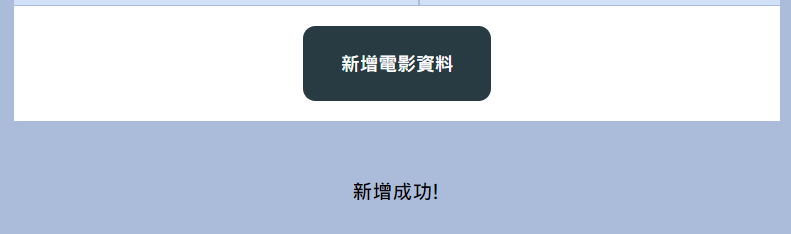
如果導演名字不存在的話，則會跳出錯誤訊息





全部輸入正確的話就能夠成功加入資料(若電影已存在會擋掉)





我們可以用電影檢查看看，可看出確實加入成功



從導演這邊做檢索也看出這個導演多執導了一部電影



1. **演員:**

可根據圖中共4個參數進行調整，搜索後會給予符合所有條件的表格。

除了會提供演員的基本資訊外，會提供他演了哪些電影



例如我想尋找日本的男演員



也可以新增演員的資料



如果演員已存在則會擋掉



1. **導演:**

可根據圖中共4個參數進行調整，搜索後會給予符合所有條件的表格。

除了導演的基本資訊外，會提供他執導了哪些電影



例如我想搜尋名字有Sam且有執導過蜘蛛人相關電影的Marvel男導演



若我將該導演刪除，則這個導演執導的電影也會跟著不見，包括相關castings，獎項，與評分等等

可發現spider man 1,2,3都找不到了



我們也可新增一個導演



若工作室不存在則會跳出通知



若導演已存在也會有通知



1. **工作室:**

可根據圖中共3個參數進行調整，搜索後會給予符合所有條件的表格。

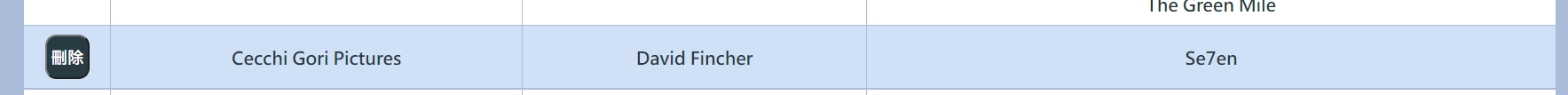
(如果沒有輸入的話則會使用預設值)

****

例如我想要尋找製作出The Avengers的工作室，會輸出該工作室名稱，有哪些導演所屬這個工作室，以及這個工作室還製作了哪些電影

若把這個工作室刪除的話，其列表的導演與電影都會一同刪除(包括評分，獎項，castings)

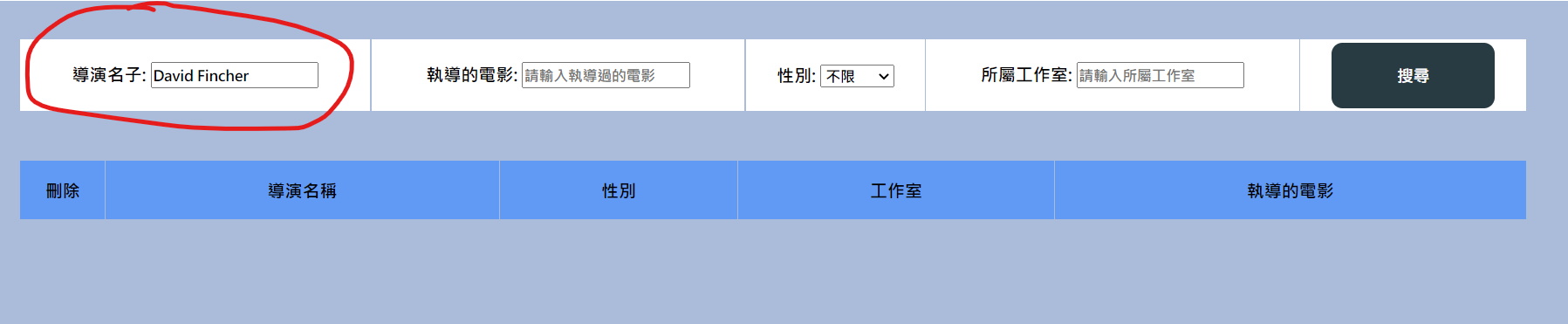
例如我將**Cecchi Gori Pictures**這個工作室刪除



movies中搜索Se7en已找不到



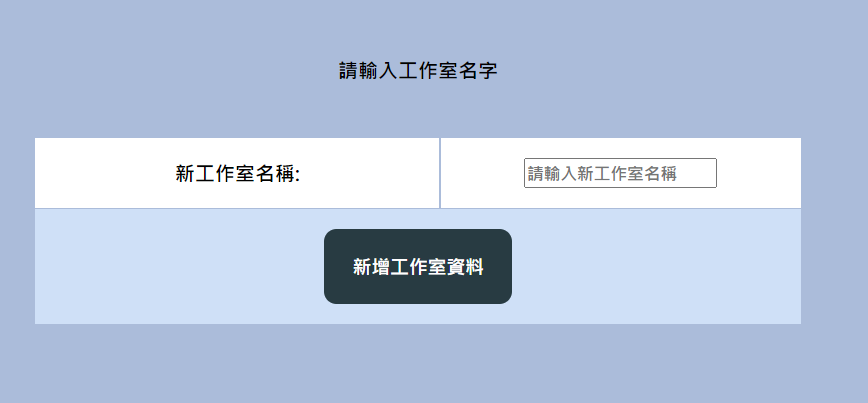
導演中搜索David Fincher也找不到了



我們也可以新增一個工作室



若沒輸入會有警告



若輸入一個已存在的工作室也有警告

****

1. **評論:**

可根據圖中共3個參數進行調整，搜索後會給予符合所有條件的表格。

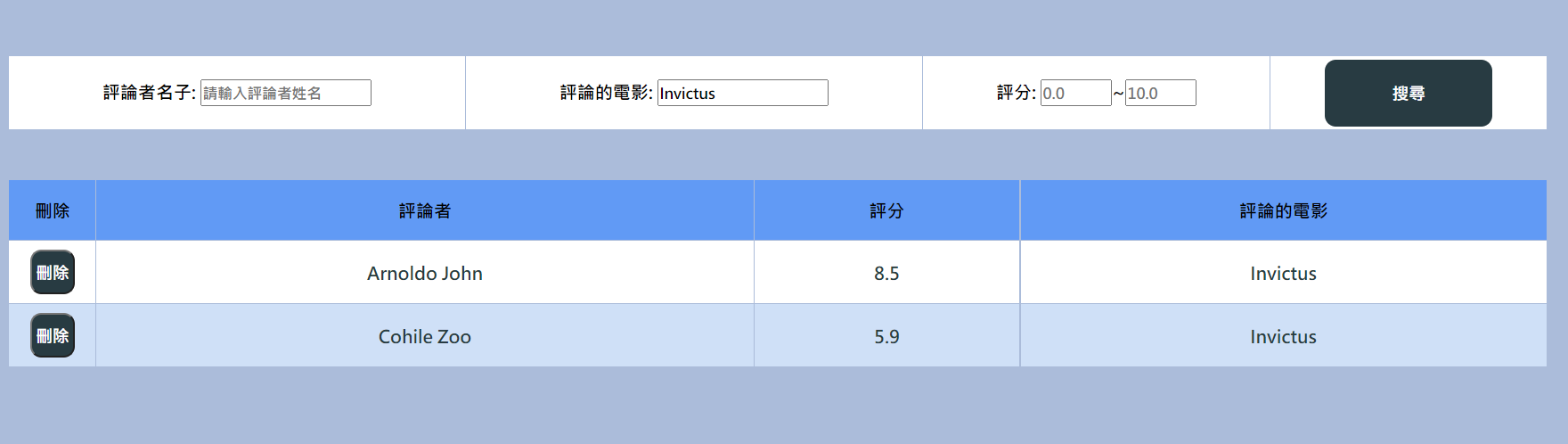
(如果沒有輸入的話則會使用預設值)



例如我想搜尋電影Invictus的評論，並且需要6分以下，會返回評論者，評分，以及評論的電影



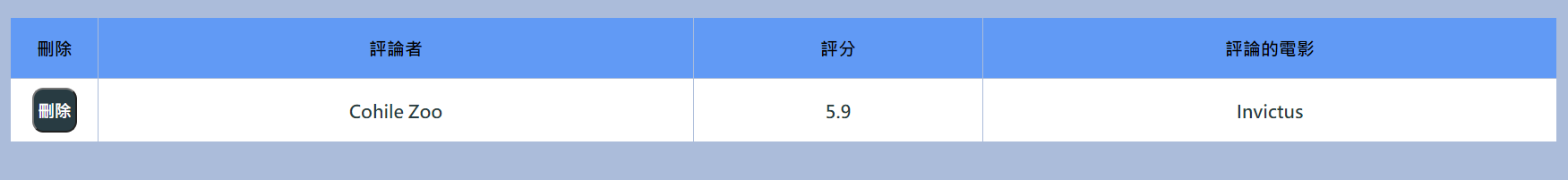
我也可以將這個評論刪掉，在刪掉之前我們先來看電影Invictus的所有評分，可發現總共有2個



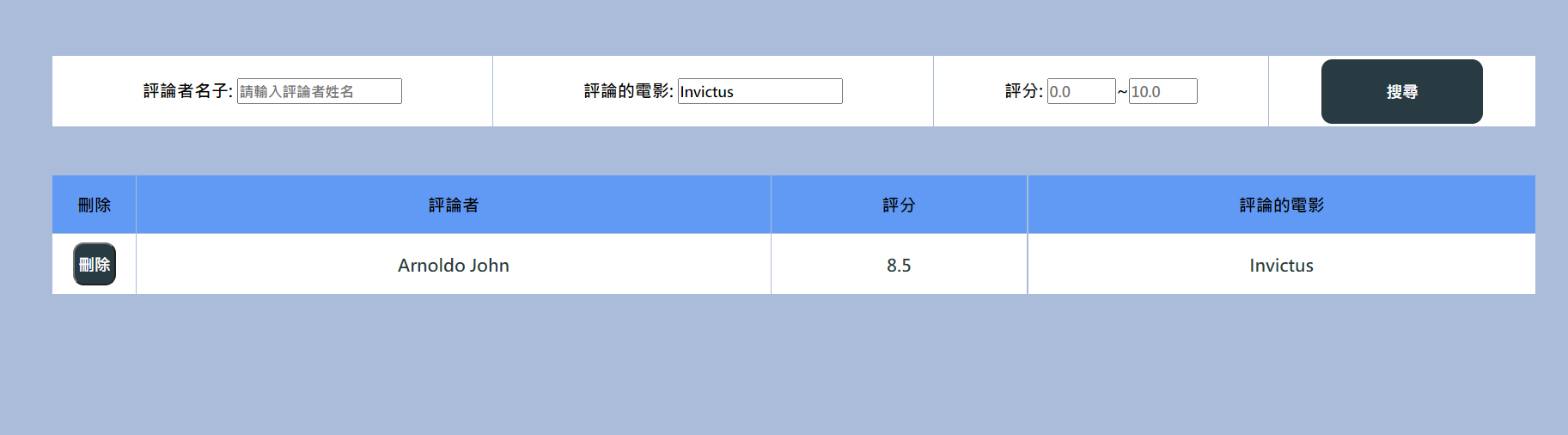
若用movies去查詢他的評分的話，可看到查詢結果為7.2分，是(8.5+5.9)/2



所以如果我將下面這個評論刪掉



可看到電影Invictus的評論只剩一個



去movies搜尋也變成了8.5分



我們也可以新增評論，一樣有做防呆機制



假如說我對電影Invictus新增一個2.6分的評論

****

在查詢時成功看到新增評論

****

假如我直接在movies中把電影Invictus刪除掉

****

那麼我在評論搜尋電影Invictus就找不到相關的評論了，因為電影已經不存在



**6.Castings:**

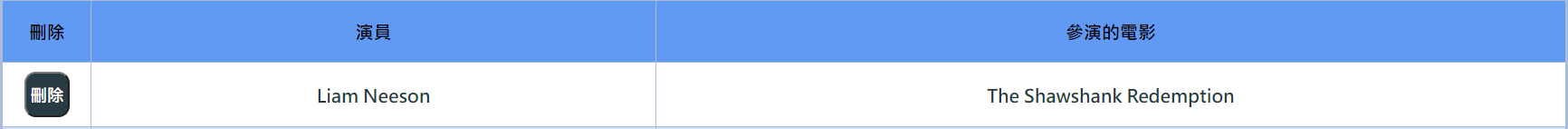
可根據圖中共2個參數進行調整，搜索後會給予符合所有條件的表格。

Castings代表的是哪個演員有參演哪部電影，因此在movies中去搜索特定的演員就是看這個table



例如我想要查詢The Shawshank Redemption這部電影有誰參演



我們先來看一下這列

如果我用Liam Neeson是能夠找到The Shawshank Redemption這部電影的



但如果我將這筆casting資料刪除的話，我們先看castings的檢索結果

可發現Liam Neeson已經不見了



我們再看movies的檢索結果

利用Liam Neeson就不能找到The Shawshank Redemption這部電影了



當然我們也能夠新增casting資料

會去偵測演員是否存在

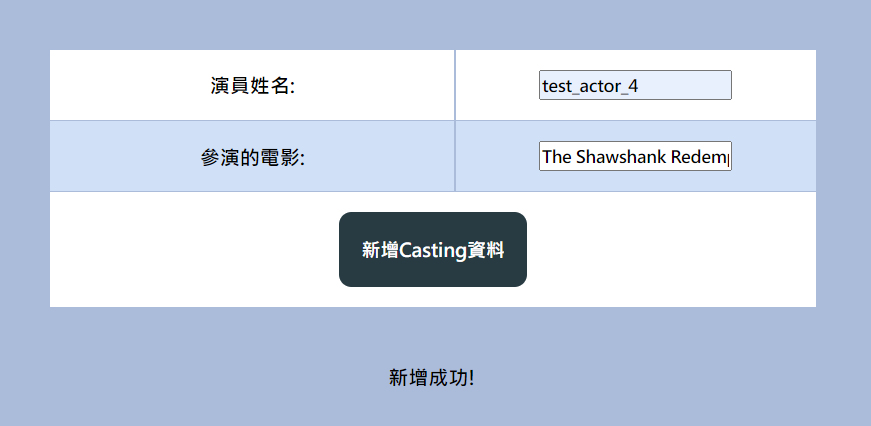


也會偵測電影是否存在



都存在的情況之下才會新增castings

假如我們新增一筆test\_actor\_4參演The Shawshank Redemption的資料



之後透過test\_actor\_4就能找到The Shawshank Redemption這部電影



**7.獎項**

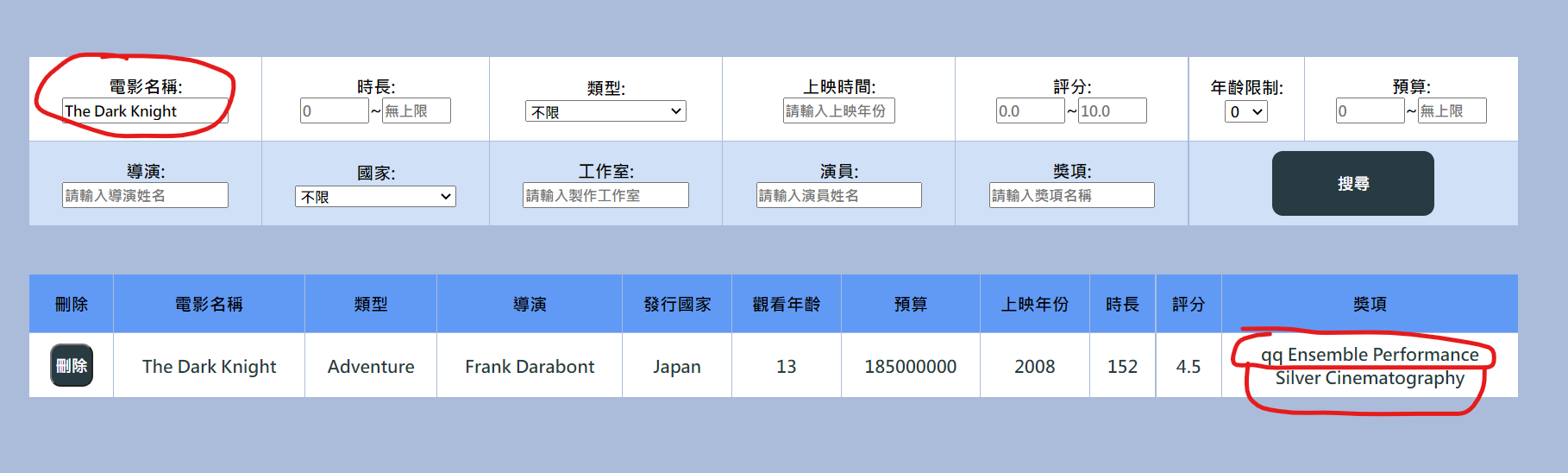
可根據圖中共個參數進行調整，搜索後會給予符合所有條件的表格。



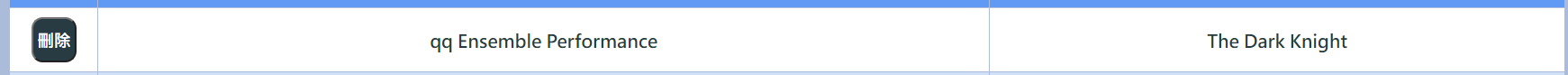
例如我想看電影The dark Knight有哪些獎項



如果我們用movies去做檢索的話一樣會有這些獎項



與movies不同的是，Award的主體是獎項，所以我們將下面那列刪除掉



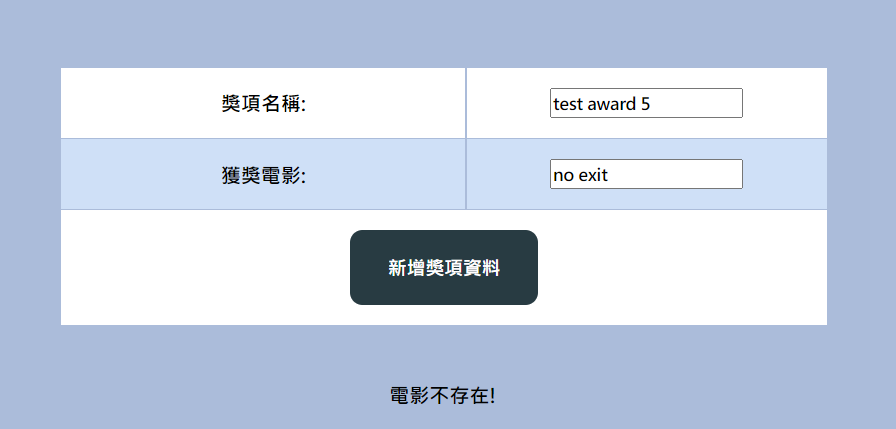
再進行一次The dark Knight有哪些獎項的檢索，會發現上面的獎項不見了



我們再換到movies去做查詢，可看到他所列出來的獎項也少了一個



我們一樣可以對獎項去做新增，一樣有防呆機制，電影不存在的話會擋掉



我們將The dark Knight新增一個名為test award 5的獎項



進行檢索，可看到新增成功



登入系統；



進入我們的系統之前，會要求登入，如果是root等管理者的話才能夠新增與除除資料，其他使用者只能夠進行資料的檢索

一般使用者畫面，沒有刪除與新增功能

