

Final Project Report

109550094 陳侑澤

1. Motivations

高中在選擇大學時，往往需要同時考慮各種因素，除了校系外，還需要面對各種篩選標準，網路上雖然有一些協助工具，但往往也都差強人意。個人高中時還要用 excel 一個一個進行比對，相當之麻煩，因此我決定針對此部分製作一個小工具，並根據我高中當初實際需要的功能進行製作。大學管道有繁星與申請兩種管道，個人是走繁星管道，對此管道規則較熟，因此選擇這部分製作。

2. Application description

這個程式可提供歷年繁星推薦的各校系查詢，並提供許多不同的篩選方法，方便進行各種交叉比較。同時可輸入學測成績，自動過濾分數不足無法錄取的學校。

3. Data sources and how you collect and import the data (manually or automatically)

資料部分，總共會需要以下幾種資料：錄取標準、校系列表、學校列表、級分五標換算表。因為這些資料都是每年才更新一次，更新頻率低，因此我採用手動更新的方式。

錄取標準、校系列表部分，在大學甄選委員會-繁星推薦

(https://www.cac.edu.tw/star111/history_statistics.php)網站，可找到各校歷年錄取標準(如下圖)，但檔

案是 pdf 檔，因此後續要額外方法從檔案中提取出相關訊息。最後的流程如下：首先以爬蟲方式，從網站下載所有學校的 pdf 錄取資料，接著用 tabula 模組，擷取出 pdf 內資料，同時進行整理(例如：五標由文字改成 0~5 表示)，同時取出所有出現過的校系，作為校系列表，最後將整了的完成的資料做成 csv，方便後續進行匯入。

學校列表的部分，一樣直接從上述網站抓出所有有參加的學校。

級分五標的部分，由於資料簡單(如右圖)，直接手動複製並轉為 csv。

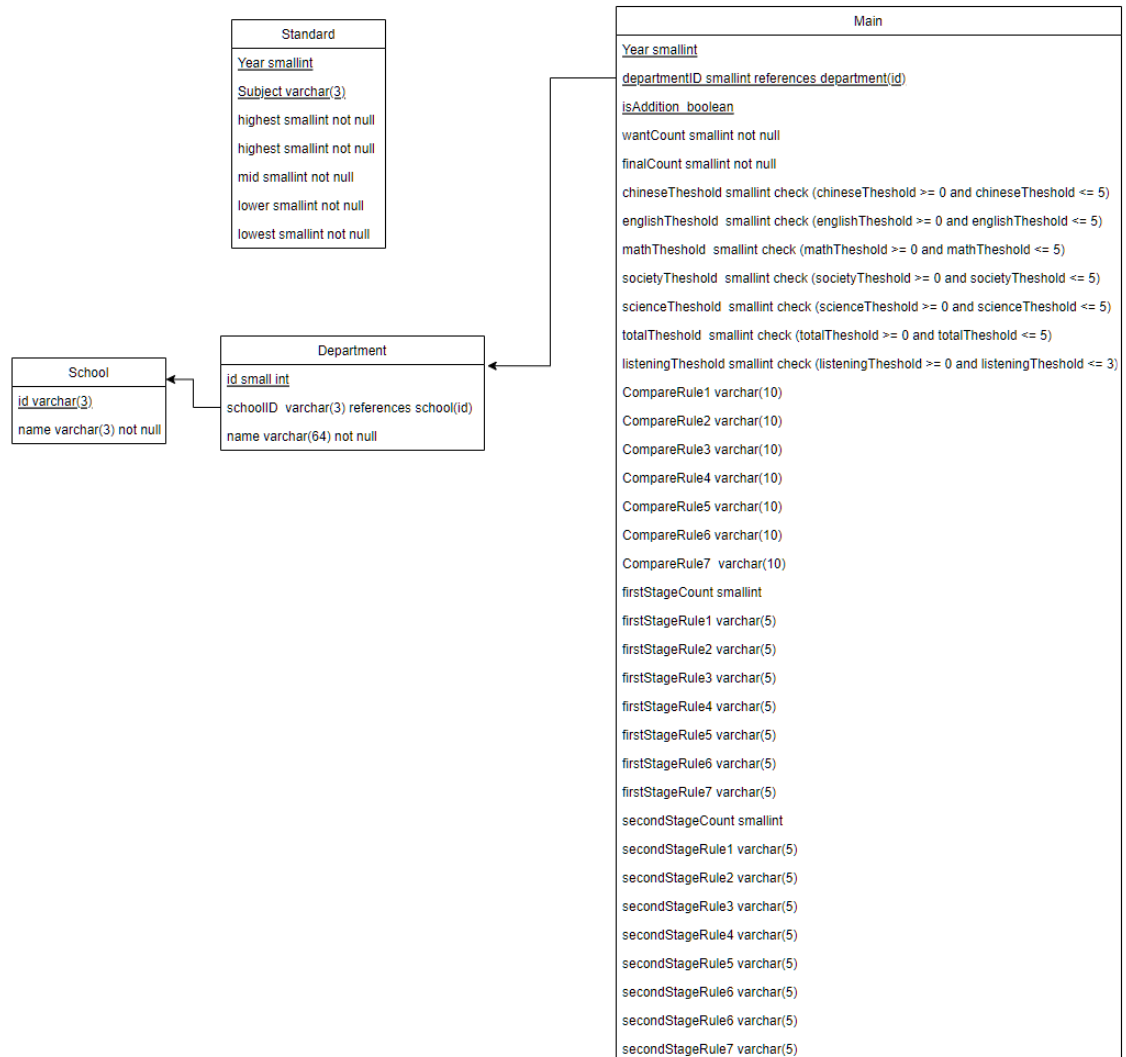
110學年度大學繁星推薦入學招生 - 第一類學群至第七類學群各校系錄取標準一覽表											
校名：001 國立臺灣大學											
校系代碼	校系名稱	招生名額	備 註	學測 - 學數		學科考試		分發志序			
				科目	備註標準及分數	項目	備註標準及分數	項目	第一類 組別 人數	第二類 組別 人數	第三類 組別 人數
00101	中國文學系	5	0	國文	滿級	112級分	國文	滿級	11	11	11
				英文	滿級	112級分	英文	滿級	11	11	11
				數學	滿級	112級分	數學	滿級	11	11	11
				社會	滿級	112級分	社會	滿級	11	11	11
				自然	滿級	112級分	自然	滿級	11	11	11
00102	外國文學系	10	10	國文	滿級	112級分	國文	滿級	11	11	11
				英文	滿級	112級分	英文	滿級	11	11	11
				數學	滿級	112級分	數學	滿級	11	11	11
				社會	滿級	112級分	社會	滿級	11	11	11
				自然	滿級	112級分	自然	滿級	11	11	11

國文	13	12	11	9	8
英文	13	12	8	5	4
數學	11	9	6	4	3
社會	13	12	10	8	7
自然	13	12	8	6	5

4. Database schema (you can use the visualization tool in DBMS directly, and list constraints that are not in the figure)

table 關係圖如下，其中所有連結皆符合 3NF。

Main table 部分，以 departmentId 為 index，以加快科系的比對。



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

可直接開啟 search.py 檔，依次輸入各項成績，如左圖，其中學校與科系可同時以關鍵字方式並可以同時搜尋多個學校，這是我認為網路上相關工具最缺乏的。

```

請輸入國文級分:15
請輸入英文級分:15
請輸入數學級分:15
請輸入社會級分:15
請輸入自然級分:15
請輸入英聽等第(A/B/C/F):A
請輸入校排%:1
是否顯示外加名額(Y/N)?
請輸入學校關鍵字(多所學校以','分隔):交大
請輸入科系關鍵字(多個科系以','分隔):資工

```

輸入完成後，會先進行幾次 query，將輸入的原始級分轉換為五標 (0~5)，如右圖。

接著會透過輸入的條件，先以 python 組成適當的 sql 指令，再傳入資料進行搜尋，相關指令如下圖。

```
1 cursor.execute(f"select * from standard where year = {year} and subject = '國文'")
2 chineseStd = cursor.fetchall()[0][2:]
3 chineseRank = getRank(chineseStd, chinese)
4
5 cursor.execute(f"select * from standard where year = {year} and subject = '英文'")
6 englishStd = cursor.fetchall()[0][2:]
7 englishRank = getRank(englishStd, english)
8
9 cursor.execute(f"select * from standard where year = {year} and subject = '數學'")
10 mathStd = cursor.fetchall()[0][2:]
11 mathRank = getRank(mathStd, math)
12
13 cursor.execute(f"select * from standard where year = {year} and subject = '社會'")
14 societyStd = cursor.fetchall()[0][2:]
15 societyRank = getRank(societyStd, society)
16
17 cursor.execute(f"select * from standard where year = {year} and subject = '自然'")
18 scienceStd = cursor.fetchall()[0][2:]
19 scienceRank = getRank(scienceStd, science)
20
21 if year < 108:
22     cursor.execute(f"select * from standard where year = {year} and subject = '總級分'")
23     totalStd = cursor.fetchall()[0][2:]
24     totalRank = getRank(totalStd, total)
25 else:
26     totalRank = 0
```

```
1 query = ('select '
2 'year, school.name, main.name, isaddition, wantcount, finalcount,'
3 'chineseThreshold, englishThreshold, mathThreshold, scienceThreshold, societyThreshold, totalThreshold, listeningThreshold,'
4 'CompareRule1, CompareRule2, CompareRule3, CompareRule4, CompareRule5, CompareRule6, CompareRule7,'
5 + stageRow +\
6 ' from('
7 ' select year, schoolId, name, isaddition, wantcount, finalcount,'
8 ' chineseThreshold, englishThreshold, mathThreshold, scienceThreshold, societyThreshold, totalThreshold, listeningThreshold,'
9 ' CompareRule1, CompareRule2, CompareRule3, CompareRule4, CompareRule5, CompareRule6, CompareRule7,'
10 + stageRow +\
11 ' from main '
12 ' join'
13 ' (select * from department) as department '
14 ' on main.departmentId = department.id '
15 f' where chineseThreshold <= {chineseRank} and englishThreshold <= {englishRank} '
16 f' and mathThreshold <= {mathRank} and scienceThreshold <= {scienceRank} and societyThreshold <= {scienceRank} and totalThreshold <= {totalRank} '
17 f' and year = {year} '
18 + additionConstraint + departmentConstraint + ') as main '
19 'join'
20 '(select * from school) as school '
21 'on school.id = main.schoolId '
22 + schoolConstraint +\
23 ' order by school.name')
```

先由最內層 query 看起，7~10 行為選擇輸出的 column，其中第 10 行會視使用者選擇不同階段而有不同的 column，接著 12~14 行 join department table 來取得系的名字，15~17 則填入使用者輸入之成績，以確保結果皆是符合使用者的科系，18 行則為篩選使用者指定的關鍵字科系，使用者輸入的每個字頭尾與中間皆會自動加入'%'，以達到關鍵字的目的。最後 19~23 再與 school table 進行 join 來取得學校名，最後以校名進行排序，使輸出較為整齊。

可看到結果輸出了所有關鍵字有「交大」、「資工」的資料。

(110)國立交通大學-資訊工程學系	26/26	前	前	頂	均	x	x	x	在校學業(3%)數學學業(2%)學測數學(12)學測自然(15)
(109)國立交通大學-資訊工程學系	26/26	前	前	頂	均	x	x	x	在校學業(3%)數學學業(3%)
(108)國立交通大學-資訊工程學系	22/22	前	前	頂	均	x	x	x	在校學業(8%)數學學業(12%)
(107)國立交通大學-資訊工程學系資電工程組	5/5	前	前	頂	均	x	x	x	在校學業(2%)數學學業(1%)學測數學(15)
(107)國立交通大學-資訊工程學系網路與多媒體工程組	8/8	前	前	頂	均	x	x	x	在校學業(2%)數學學業(1%)學測數學(14)
(107)國立交通大學-資訊工程學系資電工程組	5/5	前	前	頂	均	x	x	x	在校學業(1%)
(106)國立交通大學-資訊工程學系資電工程組	8/8	前	前	頂	均	x	x	x	在校學業(2%)數學學業(3%)
(106)國立交通大學-資訊工程學系網路與多媒體工程組	8/8	前	前	頂	均	x	x	x	在校學業(3%)數學學業(5%)
(105)國立交通大學-資訊工程學系資電工程組	5/5	前	前	頂	均	x	x	x	在校學業(2%)數學學業(5%)
(105)國立交通大學-資訊工程學系資訊工程組	8/8	前	前	頂	均	x	x	x	在校學業(2%)數學學業(6%)
(105)國立交通大學-資訊工程學系網路與多媒體工程組	8/8	前	前	頂	均	x	x	x	在校學業(5%)數學學業(3%)
(104)國立交通大學-資訊工程學系資電工程組	5/5	前	前	頂	均	x	x	x	在校學業(4%)
(104)國立交通大學-資訊工程學系資電工程組	8/8	前	前	頂	均	x	x	x	在校學業(10%)數學學業(9%)
(104)國立交通大學-資訊工程學系網路與多媒體工程組	8/8	前	前	頂	均	x	x	x	在校學業(13%)

主程式檔：

search.py - 最終查詢程式，可向資料庫查詢並輸出結果

其他程式檔：

download.py - 負責下載錄取標準 pdf 檔

main.py - 將取得的 pdf 檔提取資料後進行格式整理，輸出成 csv

convert.py - 將 main.py 中每年的 csv 整合成一個 csv

standard.py - 將各年五標輸出為 csv

standardCombine.py - 將各年五標 csv 檔整合成一個 csv

getSchool.py - 取得學校列表

createTable.sql - 資料庫表格的創建