初賽題目與解答說明

此文件只包含 Apache Pig 程式碼，Apache Hive 程式碼可以參考文步驟說明進行改寫，本次競賽語法多為基礎語法，故與SQL語法並沒有相差太遠。

詞彙說明與注意事項

1. 在本次競賽中，不考慮歌手名稱相同的問題，請直接視為同一位歌手。
2. 點播總次數: 使用者點播A歌曲10次，點播B歌曲10次，該使用者的點播總次數為20次。
3. 被點播總次數: 某一歌曲被A使用者點播10次，被B使用者點播10次，該首音樂被點播總次數為20次。
4. 個別聽眾總數: 某一歌曲被A使用者點播10次，被B使用者點播10次，該首音樂個別聽眾總數為2次。

資料格式說明，均以 TAB 作為分隔符號

userprofile.tsv (使用者個人資料)

欄位名稱	描述	型態
userid	使用者 ID	chararray
gender	性別	chararray
age	年齡	int
country	國家	chararray
registered	註冊日期	chararray

userdemand.tsv (使用者點聽紀錄)

欄位名稱	描述	型態
userid	使用者 ID	chararray
time	點聽時間	chararray
artname	歌手名稱	chararray
traname	歌曲名稱	chararray

第一題

資料基本統計，輸出格式如下

欄位名稱	數量
total_lines	X
unique_users	X
unuque_artists	X

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 使用 FOREACH 取出 ud 的 artname 欄位 (此步驟可以不做)

```
fud = FOREACH ud GENERATE artname;
```

3. 對 fud 全部欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到 total_lines

```
lines = GROUP fud ALL;
total_lines = FOREACH lines GENERATE 'total_lines', COUNT(fud) as totalcnt;
```

5. 對 up 全部欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到 unique_users

```
users = GROUP up ALL;
unique_users = FOREACH users GENERATE 'unique_users', COUNT(up) as cnt;
```

6. 對 fud 進行 DISTINCT，再對全部欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到 unique_artists

```
D = DISTINCT fud;
GD = GROUP D ALL;
unique_artists = FOREACH GD GENERATE 'unique_artists', COUNT(D) as acnt;
```

7. 使用 UNION 把三個結果合併

```
U = UNION total_lines, unique_users, unique_artists;
```

8. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test01;
STORE U INTO 'test01' using PigStorage('\t');
```

第二題 (1/2)

總註冊人數最多的五個國家，輸出欄位如下

> 計數一樣者，以國家名稱升冪排序

欄位名稱	數量
總註冊人數第一名的國家	X
總註冊人數第二名的國家	X
總註冊人數第三名的國家	X
總註冊人數第四名的國家	X
總註冊人數第五名的國家	X

1. 讀取 userprofile.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
```

2. 對 up 全部欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到每個國家的總註冊人數

```
country = GROUP up BY country;
C_country = FOREACH country GENERATE group as country, COUNT(up) as totalcnt;
```

3. 使用 ORDER BY 對總註冊人數降冪排序，國家名稱升冪排序，再使用 LIMIT 取得前五筆資料

```
LOD = LIMIT (ORDER C_country BY totalcnt DESC, country ASC) 5;
```

4. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test02-1;
STORE LOD INTO 'test02-1' using PigStorage('\t');
```

第二題 (2/2)

總註冊人數最少的國家，只顯示一筆，輸出欄位如下

> 計數一樣者，以國家名稱升冪排序

欄位名稱
總註冊人數最後一名的國家

1. 讀取 userprofile.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
```

2. 對 up 全部欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到每個國家的總註冊人數

```
country = GROUP up BY country;
C_country = FOREACH country GENERATE group as country, COUNT(up) as totalcnt;
```

3. 使用 ORDER BY 對總註冊人數升冪排序，國家名稱升冪排序，再使用 LIMIT 取得前一筆資料

```
LOA = LIMIT (ORDER C_country BY totalcnt ASC, country ASC) 1;
```

4. 使用 FOREACH 取出 LOA 的第一個欄位

```
FLOA = FOREACH LOA GENERATE $0;
```

5. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test02-2;
STORE FLOA INTO 'test02-2' using PigStorage('\t');
```

第三題 (1/2)

註冊時間最早的三位使用者，輸出欄位如下

> 計數一樣者，以使用者名稱升冪排序

使用者 ID	性別	年齡	國家
註冊時間最早的使用者	X	X	X
註冊時間第二早的使用者	X	X	X
註冊時間第三早的使用者	X	X	X

1. 讀取 userprofile.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
```

2. 使用 ORDER BY 對註冊時間降冪排序，使用者名稱升冪排序，再使用 LIMIT 取得前三筆資料

```
LOD = LIMIT (ORDER up BY registered DESC, userid ASC) 3;
```

3. 使用 FOREACH 取出題目要求之欄位

```
FLOD = FOREACH LOD GENERATE $0..$3;
```

4. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test03-1;
STORE FLOD INTO 'test03-1' using PigStorage('\t');
```

第三題 (2/2)

註冊時間最晚的三位使用者，輸出欄位如下

> 計數一樣者，以使用者名稱升冪排序

使用者 ID	性別	年齡	國家
註冊時間最晚的使用者	X	X	X
註冊時間第二晚的使用者	X	X	X
註冊時間第三晚的使用者	X	X	X

1. 讀取 userprofile.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
```

2. 使用 ORDER BY 對註冊時間升冪排序，使用者名稱升冪排序，再使用 LIMIT 取得前三筆資料

```
LOA = LIMIT (ORDER up BY registered ASC, userid ASC) 3;
```

3. 使用 FOREACH 取出題目要求之欄位

```
FLOA = FOREACH LOA GENERATE $0..$3;
```

4. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test03-2;
STORE FLOA INTO 'test03-2' using PigStorage('\t');
```

第四題 (1/2)

點播總次數最多的三位使用者，輸出欄位如下

> 計數一樣者，以使用者名稱升冪排序

使用者 ID	性別	年齡	國家	點播次數
點播總次數最多的使用者	X	X	X	X
點播總次數第二多的使用者	X	X	X	X
點播總次數第三多的使用者	X	X	X	X

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 針對 ud，依 userid 欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到所有使用者的點播總次數

```
gud = GROUP ud BY userid;
fgud = FOREACH gud  GENERATE group as userid, COUNT(ud) as cnt;
```

3. 針對 up，使用 FOREACH 取出題目要求的欄位

```
fup = FOREACH up  GENERATE $0..$3;
```

4. 使用 JOIN，以 userid 去合併欄位，並使用 FOREACH 重新命名 (可以不用重新命名)

```
joinPD = JOIN fgud BY userid, fup BY $0;
fj = FOREACH joinPD GENERATE $0 as userid, $3 as gender, $4 as age, $5 as country, $1 as totaldemand;
```

5. 使用 ORDER BY 對點播總次數降冪排序，使用者名稱升冪排序，再使用 LIMIT 取得前三筆資料

```
LOD = LIMIT (ORDER fj BY totaldemand DESC, userid ASC) 3;
```

6. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test04-1;
STORE LOD INTO 'test04-1' using PigStorage('\t');
```

第四題 (2/2)

點播總次數最少的三位使用者，輸出欄位如下

> 計數一樣者，以使用者名稱升冪排序

使用者 ID	性別	年齡	國家	點播次數
點播總次數最少的使用者	X	X	X	X
點播總次數第二少的使用者	X	X	X	X
點播總次數第三少的使用者	X	X	X	X

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 針對 ud，依 userid 欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到所有使用者的點播總次數

```
gud = GROUP ud BY userid;
fgud = FOREACH gud  GENERATE group as userid, COUNT(ud) as cnt;
```

3. 針對 up，使用 FOREACH 取出題目要求的欄位

```
fup = FOREACH up  GENERATE $0..$3;
```

4. 使用 JOIN，以 userid 去合併欄位，並使用 FOREACH 重新命名 (可以不用重新命名)

```
joinPD = JOIN fgud BY userid, fup BY $0;
fj = FOREACH joinPD GENERATE $0 as userid, $3 as gender, $4 as age, $5 as country, $1 as totaldemand;
```

5. 使用 ORDER BY 對點播總次數升冪排序，使用者名稱升冪排序，再使用 LIMIT 取得前三筆資料

```
LOA = LIMIT (ORDER fj BY totaldemand ASC, userid ASC) 3;
```

6. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test04-2;
STORE LOA INTO 'test04-2' using PigStorage('\t');
```

第五題 (1/2)

被點播總次數最多的歌手，輸出欄位如下

> 計數一樣者，以歌手名稱升冪排序

歌手名稱	被點播總次數
被點播總次數最多的歌手	X

1. 讀取 userdemand.tsv 檔案

```
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 針對 ud，依 artname 欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到所有歌手的被點播總次數

```
gud = GROUP ud BY artname;
fgud = FOREACH gud GENERATE group as artname, COUNT(ud) as cnt;
```

5. 使用 ORDER BY 對被點播總次數降冪排序，歌手名稱升冪排序，再使用 LIMIT 取得前一筆資料

```
LOD = LIMIT (ORDER fgud BY cnt DESC, artname ASC) 1;
```

6. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test05-1;
STORE LOD INTO 'test05-1' using PigStorage('\t');
```

第五題 (2/2)

被點播總次數最少的歌手，輸出欄位如下

> 計數一樣者，以歌手名稱升冪排序

歌手名稱	被點播總次數
被點播總次數最少的歌手	X

1. 讀取 userdemand.tsv 檔案

```
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 針對 ud，依 artname 欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到所有歌手的被點播總次數

```
gud = GROUP ud BY artname;
fgud = FOREACH gud GENERATE group as artname, COUNT(ud) as cnt;
```

5. 使用 ORDER BY 對被點播總次數升冪排序，歌手名稱升冪排序，再使用 LIMIT 取得前一筆資料

```
LOA = LIMIT (ORDER fgud BY cnt ASC, artname ASC) 1;
```

6. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test05-2;
STORE LOA INTO 'test05-2' using PigStorage('\t');
```

第六題 (1/2)

男性會員中，哪首歌曲被點播總次數最高，輸出欄位如下

> 計數一樣者，以歌手名稱升冪排序

歌手名稱	歌曲名稱	被點播總次數
被點播總次數最高的歌手	被點播總次數最高的歌曲	X

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 使用 FOREACH 取出 ud, up 的必要欄位 (此步驟可以不做)

```
fud = FOREACH ud GENERATE $0 as userid, $2 as artname, $3 as traname;
fup = FOREACH up GENERATE $0 as userid, $1 as gender;
```

3. 使用 FILTER 篩選出所有男性會員，再使用 JOIN，以 userid 去合併欄位

```
upm = FILTER fup BY (gender == 'm');
m_joinPD = JOIN upm BY userid, fud BY $0;
```

4. 依 artname, traname 欄位進行 GROUP，並使用 FOREACH、FLATTEN 與 COUNT 得到男性會員中，所有歌曲的被點播總次數

```
m_g = GROUP m_joinPD BY (artname, traname);
m_gc = FOREACH m_g GENERATE flatten(group), COUNT(m_joinPD) as cnt;
```

5. 使用 ORDER BY 對被點播總次數降冪排序，歌手名稱升冪排序，再使用 LIMIT 取得前一筆資料

```
m_LOD = LIMIT (ORDER m_gc BY cnt DESC, $0 ASC) 1;
```

6. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test06-1;
STORE m_LOD INTO 'test06-1' using PigStorage('\t');
```

第六題 (2/2)

女性會員中，哪首歌曲被點播總次數最高，輸出欄位如下

> 計數一樣者，以歌手名稱升冪排序

歌手名稱	歌曲名稱	被點播總次數
被點播總次數最高的歌手	被點播總次數最高的歌曲	X

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 使用 FOREACH 取出 ud, up 的必要欄位 (此步驟可以不做)

```
fud = FOREACH ud GENERATE $0 as userid, $2 as artname, $3 as traname;
fup = FOREACH up GENERATE $0 as userid, $1 as gender;
```

3. 使用 FILTER 篩選出所有女性會員，再使用 JOIN，以 userid 去合併欄位

```
upf = FILTER fup BY (gender == 'f');
f_joinPD = JOIN upf BY userid, fud BY $0;
```

4. 依 artname, traname 欄位進行 GROUP，並使用 FOREACH、FLATTEN 與 COUNT 得到女性會員中，所有歌曲的被點播總次數

```
f_g = GROUP f_joinPD BY (artname, traname);
f_gc = FOREACH f_g GENERATE flatten(group), COUNT(f_joinPD) as cnt;
```

5. 使用 ORDER BY 對被點播總次數降冪排序，歌手名稱升冪排序，再使用 LIMIT 取得前一筆資料

```
f_LOD = LIMIT (ORDER f_gc BY cnt DESC, $0 ASC) 1;
```

6. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test06-2;
STORE f_LOD INTO 'test06-2' using PigStorage('\t');
```

第七題 (1/2)

18歲 ~ 24歲的會員中 (包括18歲與24歲)，哪首歌曲被點播總次數最高，輸出欄位如下

> 計數一樣者，以歌手名稱升冪排序

歌手名稱	歌曲名稱	被點播總次數
被點播總次數最高的歌手	被點播總次數最高的歌曲	X

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 使用 FOREACH 取出 ud, up 的必要欄位 (此步驟可以不做)

```
fud = FOREACH ud GENERATE $0 as userid, $2 as artname, $3 as traname;
fup = FOREACH up GENERATE $0 as userid, $1 as gender;
```

3. 使用 FILTER 篩選出18歲 ~ 24歲的會員，再使用 JOIN，以 userid 去合併欄位

```
upf_18 = FILTER fup BY (age >= 18) AND (age <= 24);
joinPD_18 = JOIN upf_18 BY userid, fud BY $0;
```

4. 依 artname, traname 欄位進行 GROUP，並使用 FOREACH、FLATTEN 與 COUNT 得到18歲 ~ 24歲的會員中，所有歌曲的被點播總次數

```
g_18 = GROUP joinPD_18 BY (artname, traname);
gc_18 = FOREACH g_18 GENERATE flatten(group), COUNT(joinPD_18) as cnt;
```

5. 使用 ORDER BY 對被點播總次數降冪排序，歌手名稱升冪排序，再使用 LIMIT 取得前一筆資料

```
LOD_18 = LIMIT (ORDER gc_18 BY cnt DESC, $0 ASC) 1;
```

6. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test07-1;
STORE LOD_18 INTO 'test07-1' using PigStorage('\t');
```

第七題 (2/2)

25歲 ~ 34歲的會員中 (包括25歲與34歲)，哪首歌曲被點播總次數最高，輸出欄位如下

> 計數一樣者，以歌手名稱升冪排序

歌手名稱	歌曲名稱	被點播總次數
被點播總次數最高的歌手	被點播總次數最高的歌曲	X

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 使用 FOREACH 取出 ud, up 的必要欄位 (此步驟可以不做)

```
fud = FOREACH ud GENERATE $0 as userid, $2 as artname, $3 as traname;
fup = FOREACH up GENERATE $0 as userid, $1 as gender;
```

3. 使用 FILTER 篩選出18歲 ~ 24歲的會員，再使用 JOIN，以 userid 去合併欄位

```
upf_25 = FILTER fup BY (age >= 25) AND (age <= 34);
joinPD_25 = JOIN upf_25 BY userid, fud BY $0;
```

4. 依 artname, traname 欄位進行 GROUP，並使用 FOREACH、FLATTEN 與 COUNT 得到25歲 ~ 34歲的會員中，所有歌曲的被點播總次數

```
g_25 = GROUP joinPD_25 BY (artname, traname);
gc_25 = FOREACH g_25 GENERATE flatten(group), COUNT(joinPD_25) as cnt;
```

5. 使用 ORDER BY 對被點播總次數降冪排序，歌手名稱升冪排序，再使用 LIMIT 取得前一筆資料

```
LOD_25 = LIMIT (ORDER gc_25 BY cnt DESC, $0 ASC) 1;
```

6. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test07-2;
STORE LOD_25 INTO 'test07-2' using PigStorage('\t');
```

第八題 (1/2)

美國熱門歌曲排行前5名，輸出格式如下:

> - 本題的美國只有算「United States」，「United States Minor Outlying Islands」的資料不使用。
> - 排行順序以「個別聽眾總數」為主要，「點播總次數」為次要。

順序號碼	歌曲名稱	歌手名稱	個別聽眾總數	被點播總次數
1	第一筆資料	第一筆資料	第一筆資料	第一筆資料
::	::	::	::	::
5	第五筆資料	第五筆資料	第五筆資料	第五筆資料

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 使用 FOREACH 取出 ud 的必要欄位 (此步驟可以不做)

```
fud = FOREACH ud GENERATE $0 as userid, $2 as artname, $3 as traname;
```

3. 使用 FILTER 篩選出美國 (United States) 的會員，再使用 FOREACH 取出必要欄位 (FOREACH 可以不做)

```
getUS = FILTER up BY country == 'United States';
FgetUS = FOREACH getUS GENERATE $0 as userid, $3 as country;
```

4. 使用 JOIN，以 userid 去合併欄位，再使用 FOREACH 取出必要欄位

```
joinUS = JOIN FgetUS BY userid, Fud BY userid;
FjUS = FOREACH joinUS GENERATE $2 as userid, $3 as artname, $4 as traname;
```

5. 依 artname, traname 欄位進行 GROUP，並使用 FOREACH (Nested Block)、FLATTEN、DISTINCT 與 COUNT 得到美國所有歌曲的的個別聽眾總數與被點播總次數

```
GUS = GROUP FjUS BY (traname, artname);
FGUS = FOREACH GUS {
        D = DISTINCT FjUS;
        GENERATE FLATTEN(group), COUNT(D) as unique, COUNT(FjUS) as total;
}
```

6. 使用 RANK 對個別聽眾總數降冪排序，被點播總次數降冪排序，再使用 FILTER 取得前五名

```
RFGUS = RANK FGUS BY unique DESC, total DESC;
FR_US = FILTER RFGUS BY (int)rank_FGUS < 6;
```

7. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test08-1;
STORE FR_US INTO 'test08-1' using PigStorage('\t');
```

上述做法使用 FOREACH 的 Nested Block，也可以不使用，做法參考如下

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 使用 FOREACH 取出 ud 的必要欄位 (此步驟可以不做)

```
fud = FOREACH ud GENERATE $0 as userid, $2 as artname, $3 as traname;
```

3. 使用 FILTER 篩選出美國 (United States) 的會員，再使用 FOREACH 取出必要欄位 (FOREACH 可以不做)

```
getUS = FILTER up BY country == 'United States';
FgetUS = FOREACH getUS GENERATE $0 as userid, $3 as country;
```

4. 使用 JOIN，以 userid 去合併欄位，再使用 FOREACH 取出必要欄位

```
joinUS = JOIN FgetUS BY userid, Fud BY userid;
FjUS = FOREACH joinUS GENERATE $2 as userid, $3 as artname, $4 as traname;
```

5. 針對 FjUS 使用 DISTINCT，再依 artname, traname 欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到美國所有歌曲的的個別聽眾總數

```
D = DISTINCT FjUS;
GD = GROUP D BY (traname, artname);
FGD = FOREACH GD GENERATE group.traname, group.artname, COUNT(D.userid);
```

5. 針對 FjUS 依 artname, traname 欄位進行 GROUP，並使用 FOREACH 與 COUNT 得到美國所有歌曲的的被點播總次數

```
GUS = GROUP FjUS BY (traname, artname);
FGUS = FOREACH GUS GENERATE group.traname,group.artname,COUNT(FjUS.traname);
```

6. 使用 JOIN，以 traname, artname 去合併個別聽眾總數與被點播總次數，並使用 FOREACH 取出必要欄位

```
jFF = JOIN FGD BY ($0, $1), FGUS BY ($0, $1);
FjFF = FOREACH jFF GENERATE $0, $1, $2, $5;
```

7. 使用 RANK 對個別聽眾總數降冪排序，被點播總次數降冪排序，再使用 FILTER 取得前五名

```
RFGUS = RANK FjFF BY $2 DESC, $3 DESC;
FR_US = FILTER RFGUS BY (int)rank_FjFF < 6;
```

8. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test08-1-1;
STORE FR_US INTO 'test08-1-1' using PigStorage('\t');
```

第八題 (2/2)

英國熱門歌曲排行前5名，輸出格式如下:

> - 英國為「United Kingdom」
> - 排行順序以「個別聽眾總數」為主要，「點播總次數」為次要。

順序號碼	歌曲名稱	歌手名稱	個別聽眾總數	被點播總次數
1	第一筆資料	第一筆資料	第一筆資料	第一筆資料
::	::	::	::	::
5	第五筆資料	第五筆資料	第五筆資料	第五筆資料

1. 讀取 userprofile.tsv 與 userdemand.tsv 檔案

```
up = load 'userprofile.tsv' USING PigStorage ('\t') as
(userid:chararray,gender:chararray,age:int,country:chararray,registered:chararray);
ud = load 'userdemand.tsv' USING PigStorage ('\t') as
(userid:chararray,time:chararray,artname:chararray,traname:chararray);
```

2. 使用 FOREACH 取出 ud 的必要欄位 (此步驟可以不做)

```
fud = FOREACH ud GENERATE $0 as userid, $2 as artname, $3 as traname;
```

3. 使用 FILTER 篩選出英國 (United Kingdom) 的會員，再使用 FOREACH 取出必要欄位 (FOREACH 可以不做)

```
getUK = FILTER up BY country == 'United Kingdom';
FgetUK = FOREACH getUK GENERATE $0 as userid, $3 as country;
```

4. 使用 JOIN，以 userid 去合併欄位，再使用 FOREACH 取出必要欄位

```
joinUK = JOIN FgetUK BY userid, Fud BY userid;
FjUK = FOREACH joinUK GENERATE $2 as userid, $3 as artname, $4 as traname;
```

5. 依 artname, traname 欄位進行 GROUP，並使用 FOREACH (Nested Block)、FLATTEN、DISTINCT 與 COUNT 得到美國所有歌曲的的個別聽眾總數與被點播總次數

```
GUK = GROUP FjUK BY (traname, artname);
FGUK = FOREACH GUK {
        D = DISTINCT FjUK;
        GENERATE FLATTEN(group), COUNT(D) as unique, COUNT(FjUK) as total;
}
```

6. 使用 RANK 對個別聽眾總數降冪排序，被點播總次數降冪排序，再使用 FILTER 取得前五名

```
RFGUK = RANK FGUK BY unique DESC, total DESC;
FR_UK = FILTER RFGUK BY (int)rank_FGUK < 6;
```

7. 刪除已存在儲存路徑目錄，並將結果使用分隔符號 tab 儲存

```
rmf test08-2;
STORE FR_UK INTO 'test08-2' using PigStorage('\t');
```

上述做法使用 FOREACH 的 Nested Block，也可以不使用，做法請參考第16頁。