/\* GERA OS ARQUIVOS ON THE FLY \*/

**proc** **sort** data=jose.art out=all1 nodupkey;

by location;

**run**;

/\* create aux fields \*/

**data** all1;

retain AA Album AA2 Album2 Art Title Art2 Title2;

set all1;

AA2=AA;

Album2=Album;

Art2=Art;

Title2=Title;

**run**;

/\* ARRUMA Art AND Title \*/

%***Padroniza***(all1,all2

,chave1=Art2,chave2=Title2

,chave3=AA2,chave4=Album2

,remove\_spec\_chars=**1**,permut\_Art\_title=**0**);

/\* Colocar logica aqui para OneHitWonder/Fave/Fast/Easy e Year \*/

/\* Colocar tb NiceCover se bater em AA/Album \*/

/\*

data all2;

set all2;

where art2="jane duboc";

run;

\*/

**%macro** ***roda***;

%global VA Bad\_VA1 Bad\_VA2 Score Tag\_Flag;

%global Genre1 Genre2 Genre3 Genre4 Genre5;

%let vari1=Year;

%let vari2=PlayedCount;

%let vari3=SkippedCount;

%let No\_Tags=3;

%let Tag\_Flag=0;

%do i=**1** %to &No\_Tags;

%if %***VarExist***(all2,&&vari&i)

%***then*** %let Tag\_Flag=&Tag\_Flag or not missing(New\_&&vari&i);

%end;

/\* selects \*/

%let VA=(lowcase(AA) in ("various","vários","varios","ministry of sound","soundtrack","promo only","theme","promo","artist",""));

%let Bad\_VA1=&VA and index(lowcase(genre),"nicecover")=0;

%let Bad\_VA2=0 /\*Covers>0 and &VA and index(lowcase(genre),"favo")>0\*/;

/\*,Covers<=0 or (&Bad\_VA1) or (&Bad\_VA2) or (not Max\_score and &VA) as Miss\_cover\*/

/\* ALL THE ABOVE RULES ARE CAPTURED IN THE SCORE \*/

%let Score=(10\*\*3)\*(not &VA)+(10\*\*2)\*(Covers>0)+10\*(index(lowcase(genre),"nicecover")>0);

%let Genre1=strip(Genre)

||case when sum(index(lowcase(genre),"favorite\easy")>0)=0 then %Str\_found(Favorite) end

||%Str\_found(Favorite\Easy)

||case when sum(index(lowcase(genre),"favorite")>0)=0 then %Str\_found(OneHitWonders) end;

%let Genre2=tranwrd(strip(&Genre1),"\\","\");

%let Genre3=case when substr(&Genre2,1,1)="\" then substr(&Genre2,2) else &Genre2 end;

%let Genre4=case when substr(&Genre2,length(&Genre3),1)="\" then substr(&Genre2,1,-1+length(&Genre3)) else &Genre3 end;

%let Genre5=case when &Genre4 ne Genre then &Genre4 end;

proc sql;

create table Covers as

select Key

,not Max\_score and &VA as Miss\_cover

,case when %***Both\_Pop***(AA,Album) then trim(AA)||"^"||trim(Album) end as AA\_Alb

,sum(calculated Miss\_cover)>**0**

and sum(covers>**0** and not calculated Miss\_cover)>**0** as Cover\_Flag

,(a.New\_Genre ne "" or &Tag\_Flag) as Tag\_Flag

,\*

from

(select **1** as Type,case when %***Both\_Pop***(AA2,Album2)

then trim(AA2)||"^"||trim(album2) end as Key

,Genre,"" as New\_Genre

%do i=**1** %to &No\_Tags;

%if %***VarExist***(all2,&&vari&i)

%***then*** %do; ,&&vari&i,**.** as New\_&&vari&i %end;

%end;

,**.** as Score,**.** as Dist\_score,**.** as Max\_Score,\*

from all2

union all

select **2** as Type,case when %***Both\_Pop***(art2,title2)

then trim(art2)||"^"||trim(title2) end as Key

,Genre,&Genre5 as New\_Genre

%if %***VarExist***(all2,Year)

%***then*** %do; ,Year,case when (Year>min(Year)>**1900**) or Year=**.** then min(Year) end as New\_Year %end;

%if %***VarExist***(all2,PlayedCount)

%***then*** %do; ,PlayedCount,case when PlayedCount<max(PlayedCount) then max(PlayedCount) end as New\_PlayedCount %end;

%if %***VarExist***(all2,SkippedCount)

%***then*** %do; ,SkippedCount,case when SkippedCount<max(SkippedCount) then max(SkippedCount) end as New\_SkippedCount %end;

,&Score as Score,count(distinct &Score)>**1** as Dist\_score

,calculated dist\_score and &Score=max(&Score) as Max\_Score,\*

from all2

group by Key) a

where compress(Key) ne ""

and location ne ""

and album ne "Beatles"

and index(lowcase(art),"roberto carlos")=**0**

group by Key

having Cover\_Flag or Tag\_Flag

order by Key,location,type;

quit;

**%mend**;

%***roda***;

/\*

data xxx;

set covers;

where art2="jane duboc";

run;

\*/

**proc** **freq** data=covers;

table cover\_flag tag\_flag;

**run**;

**proc** **freq** data=covers;

table cover\_flag\*covers / norow nocol nopercent ;

**run**;

/\* dupes? \*/

/\*

proc sql;

create table dupes as

select location,count(\*) as N,AA\_Alb,Type,miss\_cover,covers,\*

from covers2

group by location,key

having count(\*)>1

order by miss\_cover,location,type;

quit;

data xxx;

set covers;

where new\_genre ne "";

run;

\*/

/\* eliminates dupes \*/

**proc** **sql**;

create table covers2 as

select Year,max(New\_Year) as New\_Year

,Genre,strip(max(New\_Genre)) as New\_Genre

,max(Tag\_Flag) as Tag\_Flag,\*

from covers

group by location;

**quit**;

/\*

proc sql;

create table xxx as

select \*

from covers

where location in

(select location

from covers

where key="deborah blando^luz que ascende olhar");

quit;

\*/

/\* Freq is the Freq per album \*/

**proc** **sql**;

create table covers2 as

select count(\*) as Freq,\*

from covers2

group by Key,lowcase(AA\_Alb),covers>**0**,Miss\_cover,Cover\_Flag;

**quit**;

/\* Picks album with the max frequency \*/

**proc** **sql**;

create table covers2 as

select max(case when Cover\_Flag and not Miss\_cover and covers>**0** then Freq end) as Max\_Freq,\*

from covers2

group by Key;

**quit**;

/\* Picks a file with the most freq album \*/

%let Vari=%str(case

when covers>0 and Cover\_Flag and not Miss\_cover and Freq=Max\_Freq

then location end);

**proc** **sql**;

create table covers3 as

select &vari=max(&vari) as Pick,\*

from covers2

group by Key

having Pick or (Cover\_Flag and Miss\_cover) or Tag\_Flag;

**quit**;

/\* dupes? YES \*/

/\*

proc sql;

create table dupes as

select a.location,a.key,a.Tag\_flag,a.cover\_flag,a.covers,\*

from covers3 a

where cover\_flag

group by a.location

having count(\*)>1;

quit;

\*/

/\* if same art\_size,byte has ties for Max\_Freq, set them to Null \*/

**proc** **sql**;

create table Covers3 as

select a.Key

,case when Cover\_Flag and Miss\_cover

then strip(scan(max(case when Pick then AA\_Alb end),**1**,"^")) end as New\_AA

,case when Cover\_Flag and Miss\_cover

then strip(scan(max(case when Pick then AA\_Alb end),**2**,"^")) end as New\_Album

,\*

from covers3 a

group by a.Key

having not (AA=New\_AA and Album=New\_Album);

**quit**;

/\* REMOVER CAPAS \*/

/\* ELIMINATES DUPE files. Each file should appear only once.\*/

**proc** **sort** data=covers3

out=remover\_capa(keep=location New\_AA New\_Album AA Album covers) nodupkey;

where %***both\_pop***(New\_AA,New\_Album) and (covers>**0** or not &VA)

and index(lowcase(Genre),"nicecover")=**0**;

by AA Album location;

**run**;

/\* UPDATE TAGS \*/

**proc** **sort** data=covers3 out=Update\_Tags nodupkey;

by location;

where %***both\_pop***(New\_AA,New\_Album) or New\_Year ne **.** or New\_Genre ne "";

**run**;

/\* PARECE QUE UTF-8 NAO FUNCIONA \*/

%let playlist\_name=Update\_Tags;

%***changes\_tags***(&playlist\_name,Update\_Tags);

**data** xxx;

set update\_tags;

where index(art,"Roberto Carlos")>**0**;

**run**;

/\* DETERMINA ALBUMS ONDE ALGUMA FAIXA TEM CAPA E OUTRA NAO \*/

**proc** **sql**;

create table Covers4 as

select count(\*) as N,a.Key

,a.Miss\_Cover,a.Type,a.Covers,a.AA\_Alb

,case

when (index(lowcase(cover\_mime\_type),"jpg")>**0**) then ".jpg"

when (index(lowcase(cover\_mime\_type),"jpeg")>**0**) then ".jpg"

when (index(lowcase(cover\_mime\_type),"png")>**0**) then ".png"

when (index(lowcase(cover\_mime\_type),"bmp")>**0**) then ".bmp"

when (Covers<=**0**) then "no\_cover"

else "?" end as Ext

,substr(location,**1**,findc(location,"\","B")) as Subdir

,left(translate(Location,"\_\_",":\")) as File

,trim(propcase(tranwrd(lowcase(trim(calculated File)),".mp3","")))||trim(calculated Ext) as Image\_Ext

,max(case when Pick then trim(calculated Image\_Ext) end) as Image

,\*

from covers3 a

where cover\_flag

group by a.Key

having sum(a.Miss\_cover)>**0**

and sum(a.covers>**0** and not a.Miss\_cover)>**0** and

(a.Miss\_cover or Image\_Ext=Image)

order by a.Key,a.Miss\_Cover,a.AA\_Alb;

**quit**;

/\* I only need one file\*/

**proc** **sort** data=Covers4 out=Covers5 nodupkey;

by location;

**run**;

**proc** **sort** data=Covers5 nodupkey;

by key Miss\_Cover;

**run**;

/\* EXPORTS COVER \*/

%let playlist\_name=Copy\_art;

%***copies\_cover***(&playlist\_name,covers5);

/\* dupes? YES \*/

/\*

proc sql;

create table dupes as

select a.location,a.key,a.Tag\_flag,a.cover\_flag,a.covers,\*

from covers4 a

where cover\_flag

group by a.location

having count(distinct Miss\_cover)>1 or

count(distinct AA\_Alb)>1 or count(distinct Pick)>1;

quit;

proc sql;

create table dupes as

select a.location,a.key,a.Tag\_flag,a.cover\_flag,a.covers,\*

from covers4 a

where cover\_flag

group by a.location

having count(distinct a.key)>1;

quit;

\*/

/\*

proc sql;

create table check as

select Key

,sum(Miss\_cover) as Miss\_cover

,sum(Pick) as Pick

,sum(not Miss\_cover) as Not\_Miss\_cover

,sum(not Miss\_cover and Tag\_flag) as Not\_Miss\_cover\_Tag\_Flag

,sum(not Miss\_cover and not Tag\_flag) as Not\_Miss\_cover\_No\_Tag\_Flag

,count(\*) as N

from Covers5

group by 1;

quit;

proc freq data=check ;

table miss\_cover Pick Not\_Miss\_cover Not\_Miss\_cover\_No\_Tag\_Flag;

run;

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* This will standardize some tags \*/

/\* It's not the final goal \*/

/\*

proc sql;

create table Stdz\_album as

select case when sum(Key\_Freq=Max\_Freq)>0

then max(case when Key\_Freq=Max\_Freq then trim(AA)||"^"||trim(Album) end)

else max(trim(AA)||"^"||trim(Album)) end as Fix

,trim(scan(calculated Fix,1,"^")) as New\_AA

,trim(scan(calculated Fix,2,"^")) as New\_Album,\*

from Covers

group by Key

having count(distinct AA\_Alb)>1

and not (New\_Album=Album and New\_AA=AA);

quit;

\* THE BELOW WILL GENERATE SCRIPT TO ADD THE TRACKS \*

%let playlist\_name=AAA\_Stdz\_Album;

%changes\_tags(&playlist\_name,Stdz\_album,check\_art=false);

\*/