Trabalho Prático 1 – ISI

Integração de Sistemas de Informação

Trabalho realizado por:

Carlos Ribeiro nº16986

LESI

Introdução

Foi-nos proposto realizar um trabalho para a unidade curricular “Integração de Sistemas de Informação” com o objetivo de desenvolver e manipular processos de ETL - *Extract Transform Load*.

Objetivo do projeto

Este projeto tem como objetivo aceder a dados presentes em dois ficheiros CSV contendo informação sobre musicas do spotify, e através de vários processos de filtração, validação e organização, gerar um ficheiro outpt com a informação relevante aos ouvintes de modo organizado.

Transformação Desenvolvida

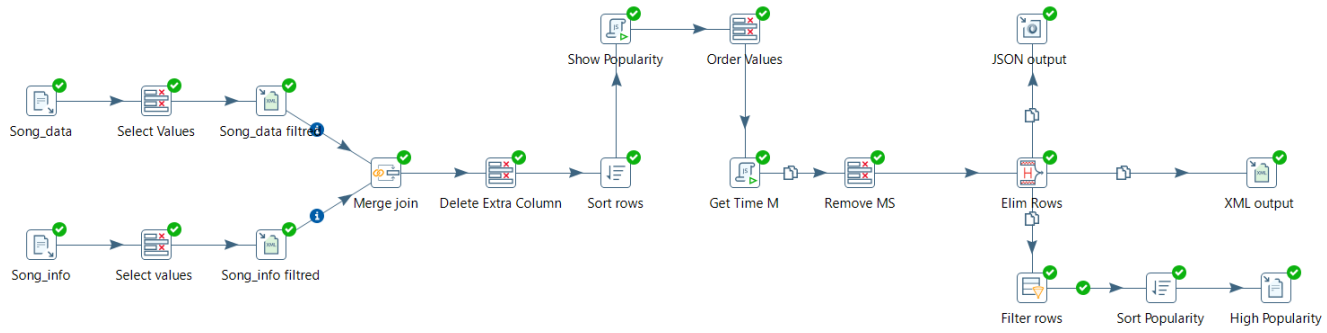

Figura 1 –Transformações do ficheiro no Kettle (TP1.ktr)

Remoção de colunas desnecessárias

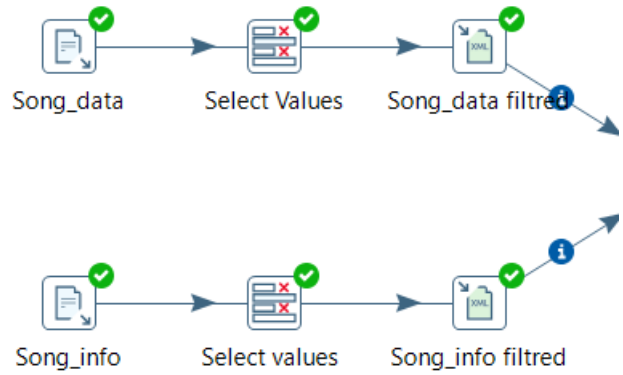

Figura 2 – Remoção de valores desnecessários (TP1.ktr)

Como os ficheiros CSV possuem informação desnecessária na forma de colunas, comecei por removê-las.

song_name	song_popularity	song_duration_ms	acousticness	danceability	energy	instrumentalness	key	liveness	loudness

song_name	song_popularity	song_duration_ms

Figura 3 e 4 – Exemplo de colunas de song_data antes e após a remoção de informação desnecessária (TP1.ktr)

Agrupamento de dois ficheiros XMl

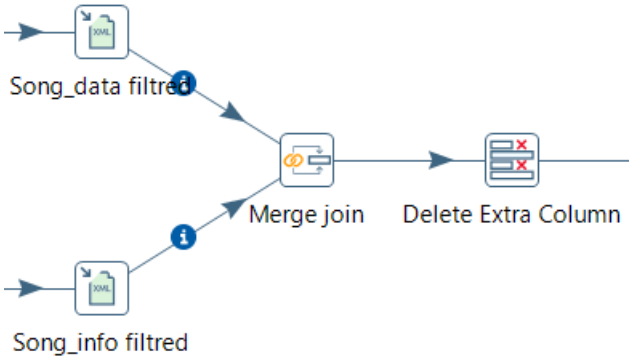

Figura 5 – Agrupamento de dois XML e remoção de coluna igual (TP1.ktr)

Após termos removido a informação desnecessária, eu queria agrupar os dois ficheiros XMl através da key *song_name* presente em ambos. Através disto posso juntar o respetivo *song_artist* presente em *song_info* ao resto da informação da música presente me *song_data*

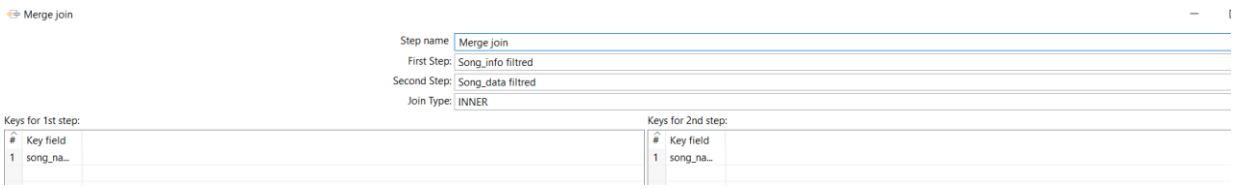

Figura 6 – Merge join utilizado (TP1.ktr)

artist_name	song_name	song_popularity	song_duration_ms
Green Day	Boulevard of Broken Dreams	73	262333
Linkin Park	In The End	66	216933
The White Stripes	Seven Nation Army	76	231733
Red Hot Chili Peppers	By The Way	74	216933
Nickelback	How You Remind Me	56	223826
Evanescence	Bring Me To Life	80	235893
Papa Roach	Last Resort	81	199893
Jet	Are You Gonna Be My Girl	76	213800
The Killers	Mr. Brightside	80	222586
Kings of Leon	Sex on Fire	81	203346
Jimmy Eat World	The Middle	78	168253
Linkin Park	Numb	63	185586
Alien Ant Farm	Smooth Criminal	75	209266
Red Hot Chili Peppers	Can't Stop	81	269000
System Of A Down	Chop Suey!	69	210240
Franz Ferdinand	Take Me Out	77	237026
blink-182	I Miss You	71	227240
Foo Fighters	Best of You	62	256600
Panic! At The Disco	I Write Sins Not Tragedies	77	187613
3 Doors Down	Kryptonite	79	233933
Thirty Seconds To Mars	The Kill (Bury Me)	69	231533
Kings of Leon	Use Somebody	79	230760
Queens of the Stone Age	No One Knows	13	255066
Caesars	Jerk It Out	62	195666
Muse	Uprising	77	304840
Plain White T's	Hey There Delilah	79	232533
Puddle Of Mudd	Blurry	28	303920
Green Day	American Idiot	78	176346
My Chemical Romance	Welcome to the Black Parade	77	311106
The All-American Rejects	Gives You Hell	71	213106
Foo Fighters	All My Life	11	262733

Figura 7 – Dados agrupados após a eliminação da coluna obtido a mais, song_name 1(TP1.ktr)

Organização de informação

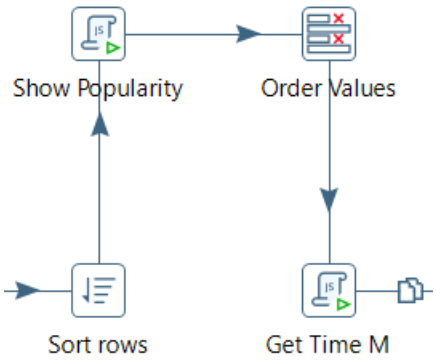

Figura 8 – Organização da informação (TP1.ktr)

Após termos a informação que queria eu organizei as linhas pelo nome do artista. Após isto criei uma coluna *popularity* através de um "modified javascript value" que verifica a coluna *song_popularity* e indica baseada nos seus valores se é *high*, *normal* ou *low*. Para além disto, transformo através de outro "modified javascript value" o tempo da duração da música, de milissegundos para minutos.

Output da informação desejada

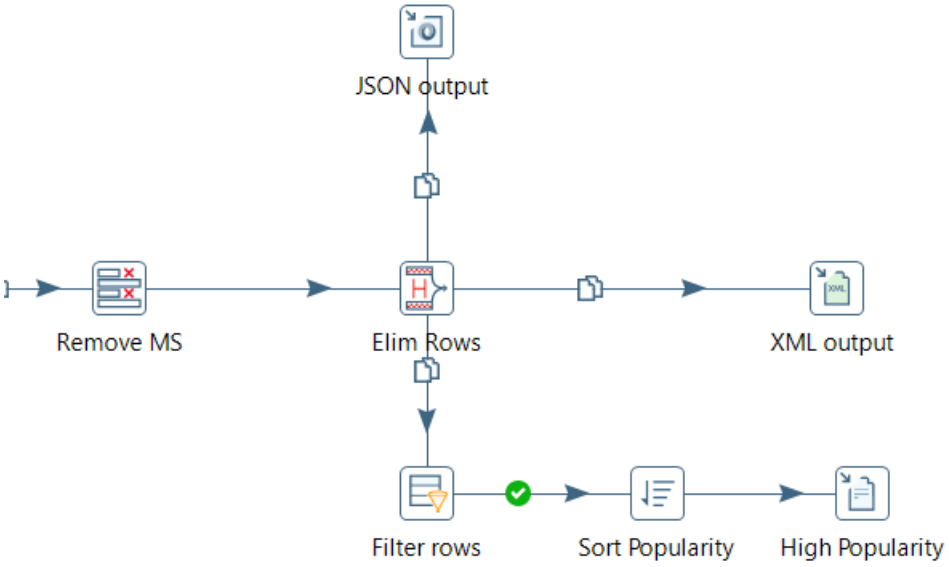

Figura 9 – Output (TP1.ktr)

Depois de obter a coluna de minutos, procedi a eliminar a coluna dos milissegundos e através de uma *Unique Rows (HashSet)* eliminei músicas que possuíssem o mesmo nome e popularidade, determinando estas desnecessárias. Feito isto, dei output a 2 ficheiros, um XML e um JSON. Para além disto, pensei em criar um ficheiro de texto, que apresentasse todas as músicas de popularidade *high* de ordem decrescente. A popularidade foi identificada através de uma expressão regular.

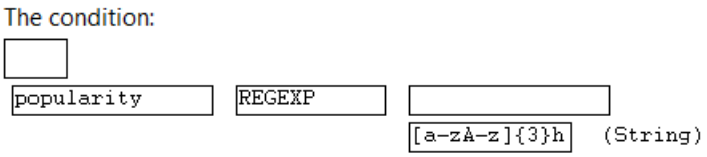

Figura 10 – Expressão regular utilizada para identificar high (TP1.ktr)

Job

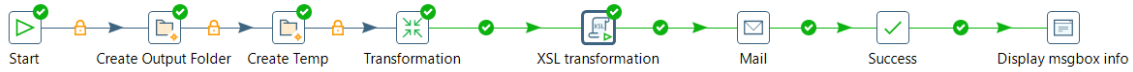

Figura 11 – Job (TP1.kjb)

No Job eu utilizei *create a folder* caso as pastas onde estão os outputs sejam apagadas, sendo de seguida utilizado a *transformation* referindo à transformação mostrada anteriormente. Eu criei um XSL que mostra, numa tabela as informações das músicas obtidas. Por fim, caso tudo seja bem-sucedido, é enviado um e-mail com uma mensagem "sucess" e é mostrado uma mensagem "Bem Sucedido" no próprio programa utilizando *Display msgbox info*.

Bibliografia

https://wiki.pentaho.com/

http://www.regexlib.com/

Material fornecido pelo professor.