**PRACTICA 3**

**ALMACÉNS E MINARÍA DE DATOS**

**Alex Baquero Domínguez**

*alex.baquero@rai.usc.es*

Tabla de contenido

[***1.*** ***Talend. Añade saída un novo campo de inciencia semanal por millón de habitantes, agrupando por ano e país ou continente*** 3](#_Toc118056029)

[***2.*** ***Apache Hop.*** 7](#_Toc118056030)

1. [***Número de muertes por país*** 7](#_Toc118056031)
2. [***Número de muertes por continente.*** 9](#_Toc118056032)
3. [***País con el mayor número de casos por semana.***](#_Toc118056033) 10
4. [***Muertes anuales por países con más de 10 millones de población.*** 12](#_Toc118056034)

## Talend. Añade saída un novo campo de inciencia semanal por millón de habitantes, agrupando por ano e país ou continente

Creamos a conexión da base de datos: COVID.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Creamos a table covid coas suas correspondientes columnas e as tablas onde almacenaremos a saida.

CREATE TABLE covid(

country varchar(40),

country\_code varchar(5),

continent varchar(10),

population bigint,

indicator\_ varchar(5),

weekly\_count int,

year smallint,

week smallint,

rate\_14\_day float,

cumulative\_count int,

source\_ varchar(60),

note varchar(40)

);

*CREATE TABLE covid\_IS\_AnoPais(*

*country varchar(35),*

*year smallint,*

*weekly\_count int*

*);*

*CREATE TABLE covid\_IS\_Continente(*

*continent varchar(9),*

*year smallint,*

*week smallint,*

*weekly\_count int,*

*incidencia float*

*);*

O job de talend quedaría desta maneira:

Mapa

Descripción generada automáticamente

Primeiramente, conectamonos a base de datos que creamos en PostgresSQL. Procedemos a copiar os datos do .csv a través do talend debido a que este na columna de year\_week ten un formato extraño. A mellor solución é mapear a táboa e dividir esta columna en duas: year e week.

*Integer.parseInt(StringHandling.LEFT(row1.year\_week,4))*

*Integer.parseInt(StringHandling.RIGHT(row1.year\_week,2))*

Una captura de pantalla de una computadora

Descripción generada automáticamente

Na inserción de datos encontramonos un problema. Algunhas columnas de tipo numérico conteñen en algunha fila o dato “NA”. Para iso, poñemos o seguinte:

*row1.weekly\_count.equals("NA")?Integer.parseInt("0"):Integer.parseInt(row1.weekly\_count)*

*row1.rate\_14\_day.equals("NA")?Float.parseFloat("0.0"):Float.parseFloat(row1.rate\_14\_day)*

*row1.cumulative\_count.equals("NA")?Integer.parseInt("0"):Integer.parseInt(row1.cumulative\_count)*

Almacenamos os datos na base de datos a través do tDBOutput\_1 e utilizamos o mesmo para redireccionar o contido as entradas tDBInput\_1 e tDBInput\_2.

Aquí os camiños da tarefa separanse depende de se queremos agrupalos por ano, país ou por continente.

* Agrupación por ano e país:
  + tFilterRow: Existen datos que son cases e deaths. Interesanos os que son cases.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* + tMap: Quedamonos coas columnas que son país, ano e casos semanales. Estas son as columnas que creamos na tabla covid\_is\_anopais

Interfaz de usuario gráfica

Descripción generada automáticamente

* + tAggregateRow: Facemos un group by por ano e pais
  + tDBOutput: Almacenamos todos os datos na táboa creada anteriormente “covid\_is\_anopais”
* Agrupación por continente.
  + tFilterRow: Quitamos os datos que conteñen o dato “NA” pois non nos interesan.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* + tMap: Quedamonos cos columnas continentes,ano,semana,casos semanales e añadimos unha nova saída: incidencias. Para calculalo, implementamos o seguinte código:

((float) row6.weekly\_count \* 1000000)/(float)row6.population

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* + tAggregateRow: Facemos un group by por continente
  + tDBOutput: Almacenamos todos os datos na tabla creada anteriormente “covid\_is\_continente”

Toda esta información está gardada en PostgreSQL, entonces facemos un commit na base de datos.

## Apache Hop.

## Número de muertes por país

Una captura de pantalla de una computadora

Descripción generada automáticamente

Importamos os datos da base de datos do postgresSQL, seleccionando as columnas que vamos a utilizar a través do table input. Despois, collemos os datos que corresponden a mortes, é decir, os que teñen un “death” (concretamente “d” na nosa base de datos):

Interfaz de usuario gráfica

Descripción generada automáticamente

Finalmente, agrupamos a consulta por países e sumamos as mortes semanales a través “group by”:

Tabla

Descripción generada automáticamente

O resultado é o seguinte:

Tabla

Descripción generada automáticamente

## Número de muertes por continente.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Importamos os datos da base de datos do postgresSQL seleccionando as columnas que vamos a utilizar a través do table input. Despois, collemos os datos que corresponden a mortes, é decir, os que teñen un “death” (concretamente “d” na nosa base de datos) e os que non teñen un código que é igual a NA (eliminamos os países que incluyen a todo un continente):

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Usamos un “group by” para agrupar a consulta por continente e sumar as mortes semanais e utilizamos un “text file output” para gardar todo nun .csv de saida.

Tabla

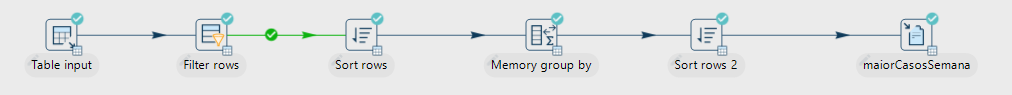
Descripción generada automáticamente

O resultado seria o seguinte:

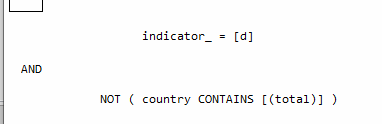
Texto

Descripción generada automáticamente

## País con el mayor número de casos por semana.

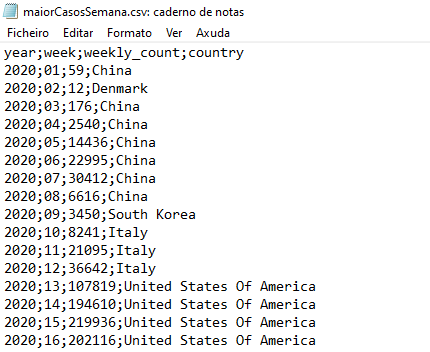


Collese a información do postgresSQL a través da Table Input e filtramos os datos de tal maneira que non haxa deaths e os a columna country non conteña un país (total).

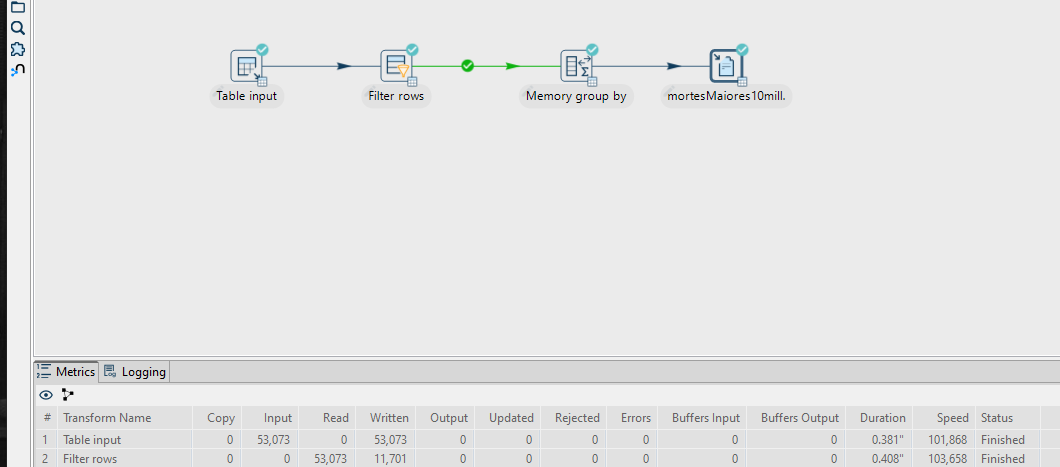


Organizamos a saída para que teñamos os primeiros países con maior casos en cada semana poñendo a year e week de forma ascendente e a weekly\_count de forma descendente. Colocamos un memory group by para agrupar cada fila pola sua data. Mandamos ao seguinte sort o country e weekly\_count xa que a primeira fila é o máximo grazas ao sort anterior.

Finalmente, imprimese nun arquivo .csv:



## Muertes anuales por países con más de 10 millones de población.



O esquema do pipeline é moi similar aos todos os anteriores. Primeiro seleccionamos as columnas das tablas que necesitamos a través do table input:

*SELECT country, population, indicator\_, weekly\_count, year FROM "public".covid\_talend*

No filter rows, poñemos da mesma maneira que se mostra nos anteriores apartados, que o indicator\_ sexa igual a d e que population sexa maior que 10 millons.

No group by, agrupamos por país e ano realizando unha operación que sume todos os datos presentes na columna weekly\_count. Finalmente almacenase nun arquivo .csvTexto

Descripción generada automáticamente