Bases de Datos Masivas (11088)
Departamento de Ciencias Básicas
TRABAJO PRÁCTICO I: Definición de Procesos ETL

Introducción:

Para la definición de procesos ETL, en primer término se solicitará al estudiante que utilice las herramientas que utiliza actualmente para luego utilizar el software Kettle/PDI de la suite Pentaho.

El objetivo de esta metodología es que el estudiante incorpore los conocimientos acerca del proceso y luego conozca las herramientas especializadas para procesos ETL. Para todas las consignas, el equipo docente proveerá las fuentes de información necesarias.

Consignas:

1. Se cuenta con el dataset Medios que cuenta con 7000 medios nacionales. Se desea normalizar esta información en una Base de Datos transaccional teniendo en cuenta que cada medio posee atributos correspondientes a su nombre, ubicación, tipo de medio y especialidad. Migre la información del archivo a una Base de Datos PostgreSQL con la siguiente estructura:

a. Medios(id,
nombre,
id_especialidad,
id_tipo_medio,
dirección,
id_ciudad),
b. Especialidades(id, descripción),
c. Tipos_medio(id, descripción),
d. Ciudades(id, nombre, id_provincia).
e. Provincias(id, nombre).

Explique someramente la metodología utilizada y estime el tiempo que le demandó la actividad.

#Primero exporto el archivo medios.xls a medios-temp.csv

#Creación de la bd

postgres=# create database medioej1 WITH encoding UTF-8;

 # Creo la tabla medios, donde voy a volcar los resultados finales, y además las tablas restantes especialidad, provincias, ciudades y tipo_medios

medioej1=# create table medio(id int, nombre varchar(50), id_especialidad int, id_tipo int, direccion varchar(30),id_ciudad integer);
CREATE TABLE
medioej1=# create table especialidades(id int, descripcion varchar (50));
CREATE TABLE
medioej1=# create table tipo_medios(id int, descripcion varchar (50));

CREATE TABLE
medioej1=# create table ciudades(id int,nombre varchar (30),id_provincia int);
CREATE TABLE
medioej1=# create table provincias(id int,nombre varchar (30));
CREATE TABLE

#Se crea una tabla temporal donde se copiaran los datos del archivo Medios que fue guardado en formato scv

mediosej1=# create temp table medios_temp (id int, medio varchar(70), provincia varchar(50), ciudad varchar(50), tipo varchar(50), especialidad varchar(50), direccion varchar(255));

mediosej1=> copy medios_temp FROM '/home/agu/Unlu/2017-2do/Masivas/tp0-1/ds/Medios-temp.csv' with DELIMITER ';';
COPY 7214

 # luego creo una secuencia para generar lo id automáticamente

mediosej1=# create sequence id_prov ;
CREATE SEQUENCE

 # asignó la secuencia a la tabla provincias (estos dos últimos pasos lo realizó para todas las tablas pero no lo vuelvo a describir)

ALTER TABLE provincias ALTER COLUMN id SET DEFAULT nextval('id_prov');
ALTER TABLE

 # luego inserto en la tabla provincia las distintas provincias que tengo en la tabla(aca no se tomó en cuenta ninguna normalización)

mediosej1=> INSERT INTO provincias (nombre) SELECT DISTINCT provincia FROM medios_temp;
INSERT 0 44

#Luego hago un inner join con la tabla ciudades y medios_temp para obtener las ciudades con los id de provincia

mediosej1=# insert into ciudades(nombre, id_provincia) (select distinct(medios_temp.ciudad) ciudad, provincias.id id_provincia from (medios_temp inner join provincias on (medios_temp.provincia = provincias.nombre)));

#cargo la tabla especialidad

mediosej1=# insert into especialidades (descripcion) select distinct especialidad from medios_temp;

#cargo la tabla tipos

mediosej1=# insert into tipo_medios (descripcion) select distinct tipo from medios_temp;
INSERT 0 21

#Por ultimo hago la unión de todas las tablas y selección de los parámetros necesarios para insertar en la tabla medios

mediosej1=# insert into medio (id, nombre, id_especialidad, id_tipo, direccion, id_ciudad)select mt.id id, mt.medio nombre, e.id id_especialidad, tipm.id_ id_tipo, mt.direccion direccion, c.id_ id_ciudad from medios_temp as mt inner join ciudades as c on c.nombre = mt.ciudad
inner join especialidades as e on e.descripcion = mt.especialidad
inner join tipo_medios as tipm on tipm.descripcion = mt.tipo;
INSERT 0 11658

2. Se cuenta con el backup de una Base de Datos de PostgreSQL, denominado 01-02-cursadas1_2003.backup , con los registros de cursada de los estudiantes de la Universidad durante el 1er Cuatrimestre 2003:Bases de Datos Masivas (11088)
Departamento de Ciencias Básicas
- Cursadas:
- Legajo,
- Plan de estudios,
- Cuatrimestre,
- Año,
- Materia,
- Condición,
- Calificación.
Por otro lado, se posee un archivo de texto plano, denominado 01-02-planes.txt con los códigos de Plan existentes y la Carrera a la cual corresponden:
- Planes y Carreras:
- Código de Plan,
- Nombre Carrera.
Para la generación de un cubo con información gerencial de menor grado de granularidad, se solicita que genere una nueva DB con una tabla

denominada rendimiento_estudiantes con los siguientes atributos:
- Legajo,
- Código de Carrera (donde el código de la Carrera es la parte entera de la división por 100 del código de Plan),
- Nombre Carrera,
- Cantidad de Cursadas,
- Cantidad de Aprobadas,
- Promedio.
Utilice el software PDI/Kettle y estime el tiempo que le demandó la actividad.
Archivo “tp1ej02.ktr” que contiene la resolución //algunos path son absolutos porque el spoon no aceptaba relativos

3. A continuación, defina conceptualmente las estructuras ETL de Kettle:
a. Transformation: Es una red de tareas logicas llamada steps, Las transformaciones son esencialmente flujos de datos .
b. Job: Los trabajos son modelos de flujo de trabajo para coordinar recursos, ejecución y dependencias de actividades de ETL.Los trabajos agregan piezas individuales de funcionalidad para implementar un proceso completo.
c. Step: Los pasos son los bloques de construcción de una transformación, por ejemplo, una entrada de archivo de texto o una salida a una tabla.Cada paso está diseñado para realizar una tarea específica
d. Hop: Los hop son rutas de datos que conectan los pasos juntos y permiten que los metadatos del esquema pasen de un paso a otro.El hop determina el flujo de datos a *través de* los pasos no necesariamente la secuencia en la que se ejecutan. Cuando se ejecuta una transformación, cada paso se inicia en su propio hilo y empuja y pasa datos.

4. Ahora, resuelva la consigna 1) con la herramienta PDI/Kettle de la suite Pentaho, a través de las transformations y Jobs necesarias para llevar adelante la solución. Tome el tiempo que demora en resolver este ejercicio con PDI/Kettle.
El ejercicio se resuelve en : tp01ej04.ktr
El ejercicio se resuelve bastante más rápido con PDI/Kettle si descartamos la curva de aprendizaje que tiene la utilización de la herramienta , también y considero mucho más importante la capacidad de extensión y modificación en spoon es mucho más poderosa

5. Cree un Job que verifique todos los días a las 14 hs si existe el archivo 01-01-medios.csv, trabajado en el punto 1), en un directorio determinado y en caso afirmativo ejecute el Job para actualizar la DB generada antes.

Este ejercicio se resuelve en:tp01ej05.kjb
6. Guarde los archivos resultantes de las actividades prácticas en una carpeta denominada tp1-<legajo> que a su vez tenga un directorio por cada uno de los puntos de este trabajo, comprima la carpeta y envíelo al equipo docente.
Referencias sugeridas:

Fuentes: https://help.pentaho.com/Documentation/
www.postgresql.org