SOLUCION A REQUERIMIENTO

Control de Cambio DWH – OCTAVIO LOPEZ BAUTISTA

Contents

[DESCRIPCION 2](#_Toc144385425)

[REQUISITOS 2](#_Toc144385426)

[SOLUCION 5](#_Toc144385426)

[EVIDENCIAS 5](#_Toc144385427)

# DESCRIPCION

Se desarrollo el ETL en “Jupyter Lab” con Python, el ETL lee los dos csv proporcionados y actualiza una tabla de dimensión y hace una carga histórica. Tambien genera un csv de salida con todos los registros que no cumplieron con las reglas de transformación.

La automatización fue realizada con el Task Scheduler de Windows para poder configurar la periodicidad al gusto.

# REQUISITOS

* AMBIENTE EN WINDOWS

Se deben de crear las siguientes carpetas para que el ETL funcione correctamente. (Nota: El usuario que vaya a ejecutar la tarea programada debe de tener todos los permisos de necesario para poder explorar las carpetas y leer los archivos para ser cargados)

A screenshot of a computer

Description automatically generated

* + ARCHIVOS\_CARGADOS:

Es la carpeta destino de los archivos una vez que fueron procesados.

* + ARCHIVOS\_EMPLOYEES:

Es la carpeta en la cual se depositará el csv con los empleados empleados nuevos o que hayan sufrido algún cambio.

* + ARCHIVOS\_RECHAZOS:

Es la carpeta en la cual se depositará el csv con todos los registros rechazados, las 3 causas de rechazo fueron: Sin salario, Sin empleador, Sin distribuidor.

* + ARCHIVOS\_SALARIES:

Es la carpeta en la cual se depositará el csv con los salarios de los empleados.

* ARCHIVOS

Los dos csv deben de cumplir con el layout acordado para que el ETL pueda trabajar con ellos. (Nota: Layouts tomado del archivo proporcionado anteriormente)

* BAT Y PYTHON SCRIPTS

Se deben depositar los siguientes scripts en una carpeta que el usuario que vaya a ejecutar tenga acceso.

A screenshot of a computer

Description automatically generated

* + ETL\_PYTHON: Este es el archivo que se deberá de configurar con el Task Scheduler para poder correr el script de Python.
  + ETL\_SOFTEK: Este archivo contiene todo el código para poder cargar a SQL Server la información.
* TASK SCHEDULER

A screenshot of a computer

Description automatically generated

Se configuro para ejecución diaria y el ejecutable es el archivo BAT, por lo que es necesario que la consola pueda ejecutar este archivo y correr archivos .ipynb

* SQL SERVER

Se levanto una instancia de SQL Server para crear la base de datos y crear las tablas necesarias para que funcione el ETL.

* + [TEST].[DIM\_EMPLOYEE]:

Esta es la tabla que se tiene que actualizar con los dos csv de “Employees.csv” y “Salaries.csv”, siguiendo las reglas de transformación.

1.- Los empleados con misma HIRE\_DATE y DATE\_ENTERED en “Employees.csv”, son empleados nuevos y deberán ser añadidos a esta tabla, buscando salarios en “Salaries.csv”.

2.-Si las fechas son diferentes, entonces el empleado ya existe en la tabla y hay que actualizarla con la nueva información, remplazando el registro viejo por el nuevo.

* + [TEST].[HISTORY\_DIM\_EMPLOYEE]:

En esta tabla se encuentra el histórico de todos los empleados en la tabla anterior para poder ver el cambio que hubo en los empleados

* + [TEST].[DIM\_EMPLOYEE\_CAMBIOS]:

Esta es una tabla intermedia donde se almacena la información para actualizar la tabla final, esta tabla se limpia con cada ejecución de ETL y su información viene de un csv llamado “Employees\_cambios”.

# A screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generated

# SOLUCION

El ETL genera 2 csv para cargar a la información a base de datos y 1 csv con registros rechazados.

* Employees\_nuevos:

Este archivo contiene todos los empleados nuevos con sus salarios y la información el formato acordado (PHONE\_NUMBER, FAX\_NUMBER, Name) y sin los registros rechazados.

Este archivo la carga directamente a DIM\_EMPLOYEE y a HISTORICO\_DIM\_EMPLOYEE, ya que no deberían de existir en la tabla DIM\_EMPLOYEE. (Solo se tienen registros de empleados nuevos o con cambios en el csv original).

* Employees\_cambios:

Este archivo contiene todos los empleados que sufrieron un cambio y deben de actualizar la tabla DIM\_EMPLOYEE, para esto primero se carga a la tabla DIM\_EMPLOYEE\_CAMBIOS y se hacen un merge para actualizar la tabla, elimando los registros existentes por los nuevos.

* RECHAZADOS:

Esta archivo contiene todos los registros rechazados junto con su causa de rechazo.

# EVIDENCIAS

* Casos de prueba:

Se cargarán los registros del archivo “Employees\_recortado.csv” con “Salaries\_Recortado.csv”, los cuales tienen:

* + Registros sin EMPLOYEE\_ID
  + Registros sin DEALERSHIP\_ID
  + Registros que no van hacer match con Salaries.csv

La tabla DIM\_EMPLOYEE contiene 5 registros, el ETL únicamente debe modificar los que cumplieron las condiciones y no modificar los registros existentes que no deben recibir cambios. (Validar conteo de registros)

En el Excel “EVIDENCIAS\_ETL” se encuentran 3 hojas:

* + ANTES\_ETL:

En esta hoja se muestra la tabla DIM\_EMPLOYEE con únicamente 5 registros y la información que hay en los dos archivos con esos EMPLOYEE\_ID, esto para ver que aunque haya salarios con esos ids , si no se encuentran en los dos archivos los ids, no se puede hacer un match y no se realizara una actualización.

* + REGISTROS\_EN\_ARCHIVOS:

En esta hoja se tiene el numero de registros del archivo de Employees, para que sea vea como se distribuyeron los registros, la suma de los 3 archivos tiene que el mismo numero de registros del archivo de entrada. (Employees)

* + DESPUES\_DE\_ETL:

En esta hoja se encuentra la evidencia de como se actualizo la tabla DIM\_EMPLOYEE con salarios nuevos para los registros que hicieron match (Salary = 1000), y también como se cargaron los empleados nuevos a DIM\_EMPLOYEE y HISTORICO\_DIM\_EMPLOYEE.

También se muestra la evidencia de RECHAZADOS.csv con sus 3 casos.