Module 5
Analítica de datos y extracción de conocimiento mediante técnicas de IA (Data Analytics)

Task 3: ETL Processes to load data into the data warehouse repository

Description:

The use case refers to a technological company which gathers data related to its orders and shipments. The company wishes to better understand the evolution of its sales and returns globally. In a more concrete way, the company is absolutely worried with improving its order's preparation time. To this aim, the company is gathering data related to the exact date when (i) the order is received, and (ii) the shipment is ready and placed in the shipment company. Apart from this critical issue, the company would like to better understand its clients, the main ordered products as well as the main returned products.

The collected data is distributed into three main datasets:

1.- Orders.

Basically, an order represents products ordered by clients. The key information gathered about the order of a product is the number of units (quantity), the sales value (sales), discount, profit, and the shipping cost of an order. The type of orders and the priority are features chosen by clients (also provided in the data set).

The main information provided for customers are their names, customer number, type of customer, zip code, city, state, country, region and market to which they belong to. This latter feature has a lot of potential as it will allow us to analyze potential markets among other factors, as well as assess which are our best clients or identify potential clients.

The main information gathered from products are the name of the product and their different categories depending on their types.

Finally, we are also interested in knowing which products have been returned or not for any reason.

2.- Returns

It contains information on which product from which region has been returned.

3.- Markets

This data set provides further information about which country belongs to which Region and to which Market. In this particular case study, this information has been provided in another different data file.

Try to help your company improve its situation by creating a data analytical repository (data warehouse) that contains the information that will allow us to contain the main data that facilitates its ulterior analytical queries. In order to create this data warehouse, this first task is to specify a multidimensional conceptual schema.

Submit a zip file that contains all the transformations defined by using the tool PDI (Pentaho Data Integration) that will populate the data warehouse schema (star schema) defined in the previous Task 2. Please, define one transformation for each dimension, one for the fact, and a job linking the execution of all the transformations. Provide all the .ktr files and the .kjb file and make a zip with all the files.

Remember that the file submitted must **include the full name(s) of the student(s) involved**.

To do:

As commented, we should create one transformation (.ktr) for each dimension and one (.ktr) for the fact. Then, we will also provide the file for the created job (.kjb). Please, take into consideration that we should populate first each dimension, and then the fact, as there are foreign keys created in all the dimensions.

Note: If you test the dimensions and fact transformations and they run properly; please, take into consideration to delete all the data from the database by using SQL developer and then, create the job (.kjb extension) and run the job.

We should choose the correct operators for each dimension considering the table attributes for the output schema (star schema) and the attributes to be populated from the provided data sets (orders, markets and returns).

Please, let us remind you to consider some of the main operators we used with populating the football schema such as join, look-up table, add sequence, and so on. Now, we will provide some tips for populating all the dimensions:

Delivery dimension:

In this dimension, we wish to have the delivery mode (*Ship_mode*) for every order. We read the orders.csv data source (input), then we only wish to have the delivery mode (select the column Ship_mode), then we need to order/sort the files (sort), then we choose unique files (within this operator we wish to avoid the orders that do not have information on the Ship_mode attribute) and for this reason we discard the files that Ship_Mode is N. Then, we add sequence (add_sequence operator), and finally we put the two main attributes in the output table (Ship_Mode and the autogenerated attribute).

Note: remember that during the lectures we pointed out that it is compulsory to sort rows before applying several operators such as join, unique files and so on.

Note 2: Please, remember what we said in previous documents that some data profiling should be done by hand before applying new techniques in the following week (e.g. to discover that there several rows are Ship_mode N).

Order Priority Dimension:

Please, note that the transformations and operators are absolutely similar to those applied in the delivery dimension. In this particular case, instead of Ship_mode, we need to gather the *Priority_Type* attribute. Then, please, use the same six operators.

Product Dimension:

As in the same previous two dimensions. We need to use the same six operators. Please, pay attention that in this particular case, we need to gather three attributes from the orders.csv to populate the product dimension table: Product name, Category and Subcategory (I mean that in contrast to the two previous dimensions, in this dimension we need to select three attributes).

Date dimension:

First, please, we need to make a data profiling to find out how many days we have in our system, i.e. how many days we have had orders ? Please, take into consideration if we also have leap-years.

First, we need to use the operator Generate rows in order to generate all the rows in the date format (with any name) to allocate all the different days in our orders.csv (from 2010 to 2017). Then, we add an autogenerate to count the days from the first registered day. Then. We start here by 0, and we increment in one. We use the calculator to add the number of days since 2010 (the autogenerated one) to the default value: Date A (generated rows) + B days (the recently created autogenerate). Then, we extract day, month and year of the date. We add sequence, and finally, we upload all the data in the output table.

Returns dimension:

What we need here is to have the reference if the order has already been returned or not. Doing this is a very easy set of operators, but new ones. Actually, take into consideration that we need to populate in the returns table that we have to rows, one (1) for returned and another one (2) for not returned. This could be done by hand very easily by using the SQL. However, by using PDI, it is also easy. First, we generate a boolean value, then add a sequence, we use a new operator (Value mapping A→B) to define that 1 is returned and 2 is not returned; we select both the generated id boolean and the new add sequence, and place in the output table.

Customer dimension:

This dimension shares several operators with the previous ones, but we need to consider one issue before. In the data sources, we have the third file markets, where we have the information on the markets. Therefore, we need to make a join with orders and markets in order to have all the info. Let's summarize it.

First, we read the input from the markets.csv file, and then we sort them. In a parallel way:
We read the input from the orders.csv file, we select ONLY the attributes from customers (customer id, name, … city,… and country), then we sort them. Then, we make a join (remember to specify a LEFT join) with the *attribute* in order to have not only the country but the markets. Then, as in the previous dimensions, we select the needed rows (this allows us to discard the repeated rows from the join operator), sort them, select unique rows (with the ID Customer), add sequence and then, output table.

Fact table (orders)

Before the set of look_up tables as we did in the lessons, we need to make a set of steps.

From returns.csv we need to select the order ID, add a constant as returned (as all the orders contained in this file has been returned), and then sort. On the other hand, from orders we need to select the Order ID, and make a join with the previous way. Then, right after the join, we need to make sure that we have the value mapping read nulls from the left join, or in other words, unreturned products, in the origin field configuration, leave it empty and destination "No returned". In doing so, we define the A→B value mappings, and if the source attribute is NULL, the output value is Unreturned. We select the attributes needed. Then, We calculate the measure "preparation_time", We use the Date A - B (in days) operation. Be careful, in this case Date A is Ship date, and Date B Order date; then extract the day, month and year for the date dimension. These two operators are the Calculator operators.

Finally, we add all the lookup tables and finally, we add sequence and make the output table.