DEPARTAMENTO DE ELETRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA
UNIVERSIDADE DE AVEIRO

EXPLORAÇÃO DE DADOS / DATA MINING

(Part I) Laboratory Assignment 03: Data warehousing design and Data extraction, transformation and loading (ETL) using PostgreSQL and Pentaho (PDI/Kettle)

Northwind is a popular and relatively simple database that captures the sales transactions between a hypothetical company, the *Northwind traders*, and its customers, as well as purchase transactions and suppliers. The database is available at https://github.com/cjlee/northwind, but it is recommended to use the instance of this database installed in our server *deti-sql-aulas.ua.pt*. The database diagram is also available at the same website. The database stores information about:

- Suppliers, customers, employee details of *Northwind traders* and product information.
- Details of the shippers who ship the products from the traders to the customers.
- Purchase order transactions between the suppliers and the company.
- Sales order transactions between the customers and the company.

This assignment aims at implementing the first steps to create a *Data Mart* for the *Northwind traders* company. The focus is on the physical database design and ETL The conceptual schema (multidimensional data model) is depicted at the end of this document. The source file (BI Modeler tool) is *Northwind.bim*.

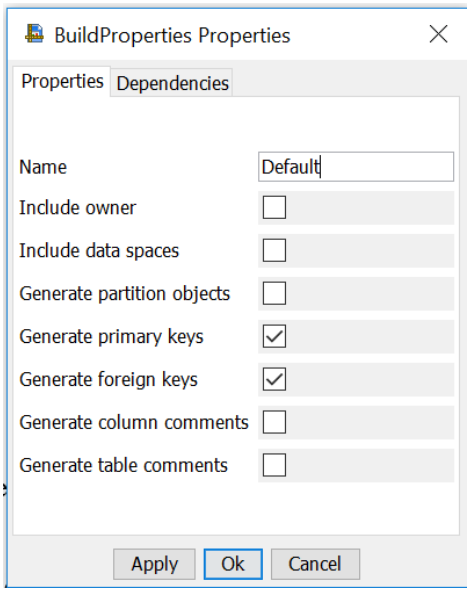
> *The conceptual schema represented using the DFM notation [1] was adapted from [2], but other solutions exist in literature.*

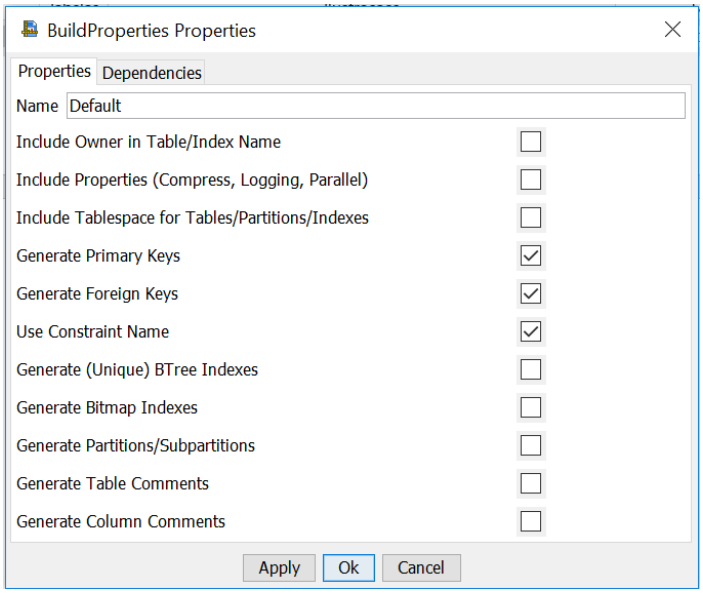
Setup

You will use the BI Modeler tool to generate the scripts to create the data mart.

- Open the *Northwind.bim* file and display the *FactSchema Sales*.
- Change the view mode to *Relational Model* and then execute the command *Relational Model → Forward engineering model*. Have a look at *Glossary Properties*. Go to *Hierarchies* and change the strategy of the *Order* dimension to *Embedded dimension*. Later in this course, you will learn about different strategies to implement hierarchies, but for now, just use the default properties for the other dimensions.
- *At this point you should drag all table in the relational model to the design pane and see the database schema generated by BI Modeler. You should define the data types and the size of the attributes in each table according to the specifications of your target DBMS, but you do not need to do it in this assignment because it is already done.*
- Execute the command *Relational Model → New Physical Model* to define the properties to be used in the creation of the data mart. If you are using Postgres, it is recommended to choose

Oracle 10g/11g in the *DBMS* combo box. Give a name to your Physical Model and then edit the Default build properties as follows (SQL Server in the left or Postgres/Oracle in the right):





- Execute the command *Generate DDL*, add all tables and generate the script to create the data mart in your target DBMS.

- Remarks: There is only a subject area. You will learn about partitions later in this course.

At this point, you are ready to create the data mart. Note that the proposed design combines normalized (snowflake schema) and non-normalized dimensions (star schema). To complete this task, you must:

- Edit the script and define an autoincrement Primary Key *in D_Location* and *D_time*. Use the clause *'identity(1,1)'* in *Microsoft SQL Server* or the data type *'serial primary key'* in *PostgreSQL*.
- Comment the primary key definition in *F_sales* to be able to test the ETL tasks requested below more easily (i.e., to avoid errors of the type 'cannot insert null values into…' when the data flow tasks are incomplete). You can uncomment this statement at the ETL package is complete but notice that defining a good primary key in a facts table is not trivial.

Exercises & questions

You must consider that you are creating the data mart for the first time, hence you are free to recreate the tables or clear their contents whenever you need. This would not be the case if you were migrating data into a data mart already in production.

1. Create a data flow task to load data into the dimensions *D_product* and *D_category*. As table *D_Product* has a foreign key to table *D_category*, the data flow task that fills dimension *Categories* must be performed before the data flow task that fills dimension *Products*.

   Tip: include a transformation to delete all records from *D_product* and *D_category* before loading new data,

2. Create a data flow task to load data into *D_location*, *D_supplier* and *D_customer*.

3. Create a job to load the facts table (considering only the dimensions referred in previous exercises).

   This task must only start after the tasks for loading the dimension tables are finished.

4. The aim of the exercise is to show how to use the Slowly Changing Dimension operator in *Kettle*. You must devise a plausible use case and create a demo (tutorial) on how to use this operator.

References

[1] Stefano Rizzi. Conceptual modeling solutions for the data warehouse. In Data Warehousing and Mining: Concepts, Methodologies, Tools, and Applications, J. Wang (Ed.), Information Science Reference, pp. 208-227, 2008.

[2] Alejandro Vaisman and Esteban Zimányi. Chapter 4 - Conceptual Data Warehouse, In Design Data Warehouse Systems, Design and Implementation. Springer-Verlag Berlin Heidelberg, p.89-119, 2014. ISBN 978-3-642-54655-6.

You must save the solutions of the exercises because you will need to submit them on November 4 (date to be confirmed).

Multidimensional data model for the Data Warehouse of the Northwind Traders company