ISIT312 Big Data Management

Extraction, Transformation, and Loading

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

Extraction, Transformation, and Loading

Outline

Extraction, Transformation, and Loading (ETL)

Extract data from internal and external sources, transform data, and load data into a data warehouse (ETL)

No agreed way to specify ETL at a conceptual level

We study conceptual ETL design

Conceptual model based on the Business Process Modeling Notation (BPMN)

- Users already familiar with BPMN do not need to learn another language to design ETL
- BPMN provides a conceptual and implementation-independent specification of processes
- Processes expressed in BPMN can be translated into executable specifications(e.g., Microsoft's Integration Services)

Extraction, Transformation, and Loading

Outline

Conceptual ETL Design using BPMN

Basic assumption for using BPMN as conceptual model: ETL process is a type of business process

There is no standard model for defining ETL processes

Each tool provides its own model, too detailed to be conceptual

Using BPMN constructs we define the most common ETL tasks and define a BPMN notation for ETL

ETL process: A combination of control and data processes

- Control processes manage the coarse-grained groups of tasks
- Data processes detail how input data are transformed and output data are produced

Two kinds of tasks in ETL conceptual modeling

- Control tasks highlight the control procedures provided by BPMN. Represent a workflow (arrows represent the precedence between activities)
- Data tasks refer to the tasks that directly manipulate data during an ETL process. Represent a data flow (arrows represent data 'flowing' along them)

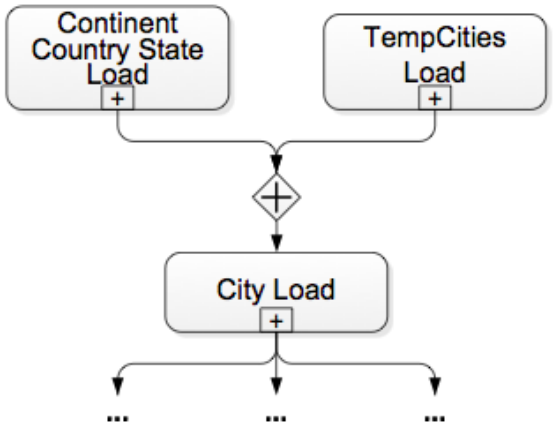
Control Tasks

Represent the workflow sequence or orchestration of the ETL process independently of the data flow

Control tasks are represented by means of BPMN constructs described

For example, gateways are used to control the sequence of activities in an ETL process

The most used types of gateways in an ETL context are exclusive and parallel



Data Tasks

Show how data are manipulated within an activity

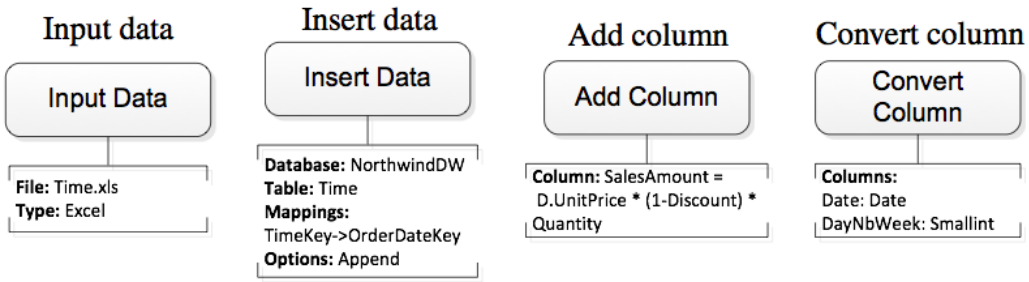
At lower abstraction level than control tasks

Represent activities typically carried out to manipulate data: input and output data, data conversion and transformation(for instance, change the data type of an attribute, add acolumn, remove duplicates, and so on)

We denote these tasks unary data tasks since they receive one input flow

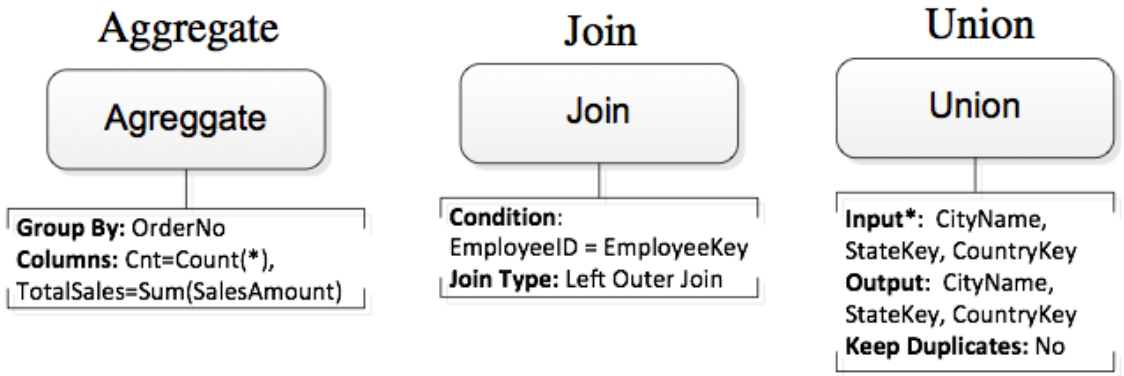
n-ary data tasks receive as input more than one flow (e.g., this is the case of union, join, dfference,...)

Row operations are the transformations applied to the source or target data on a row-by-row basis, e.g., updating the value of a column



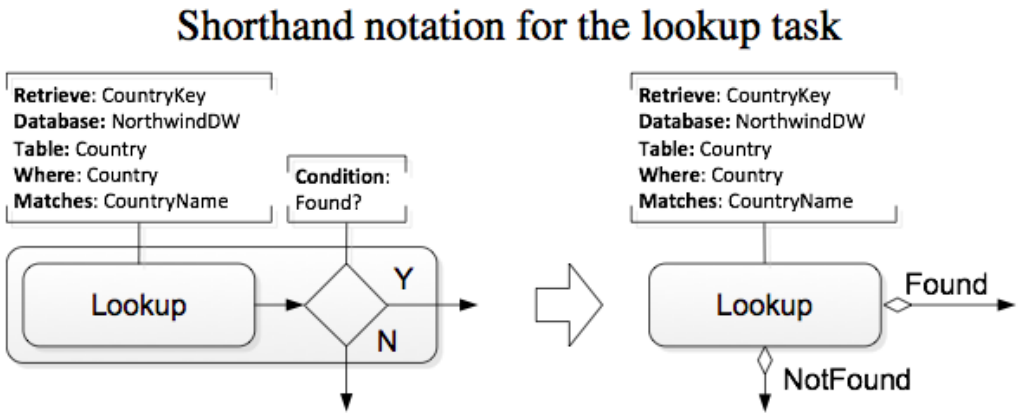
Rowset Data Tasks

Rowset operations deal with a set of rows, e.g., aggregation is a rowset operation



Lookup Data Tasks

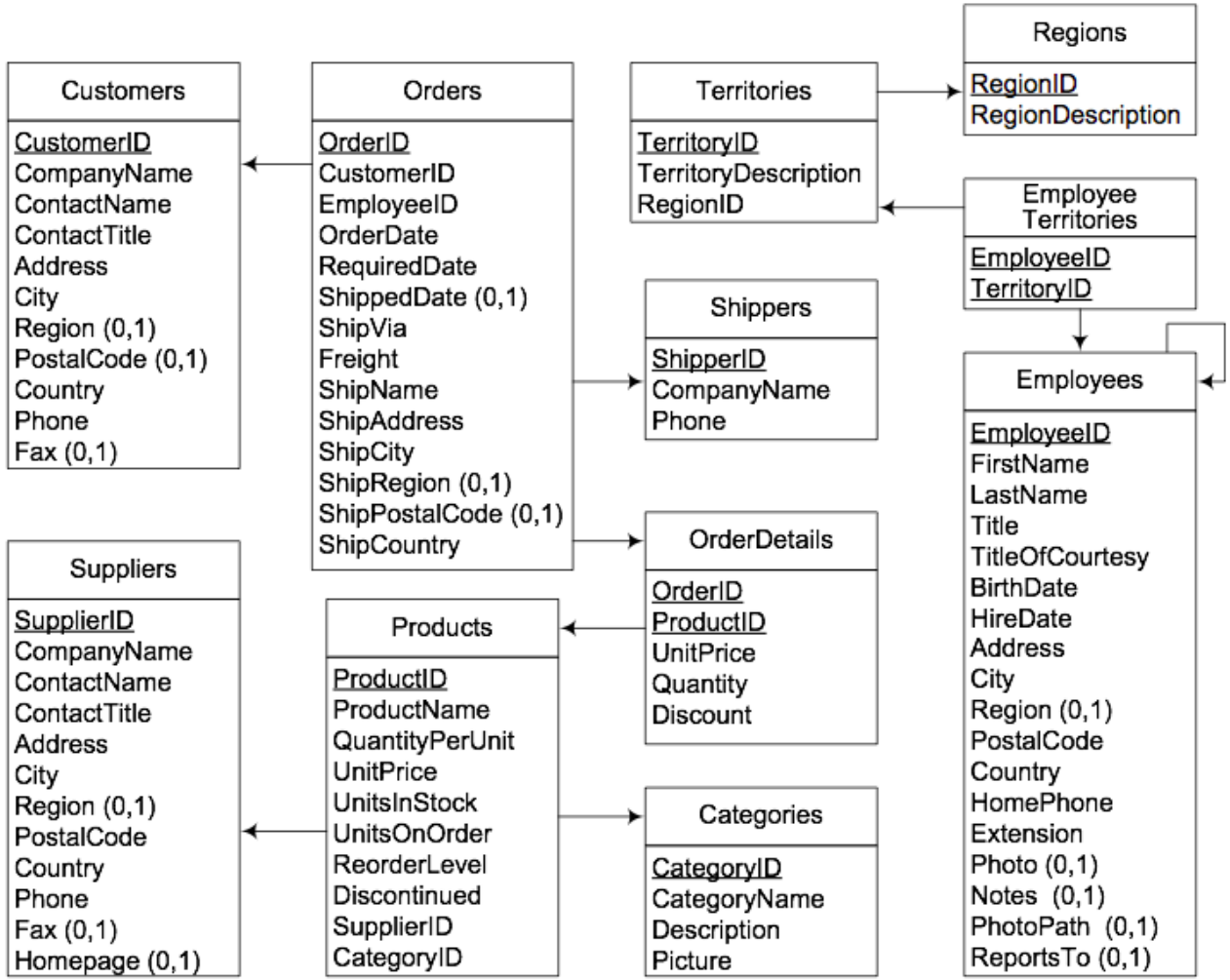
Lookup Data Tasks check if some value is present in a file. Immediately followed by an exclusive gateway with a branching condition. We use a shorthand replacing these two tasks by 2 conditional flows.



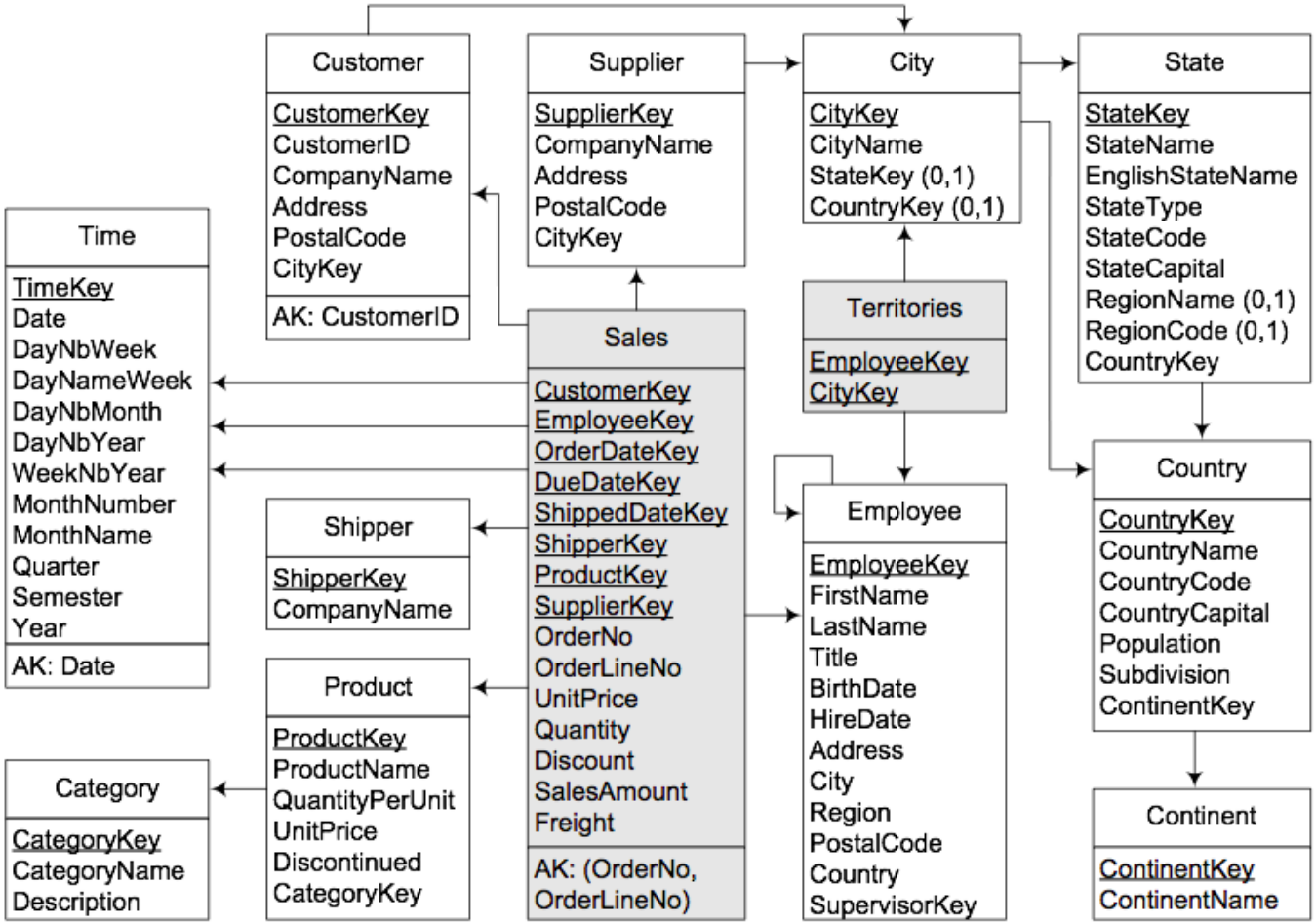
Extraction, Transformation, and Loading

Outline

Schema of the Northwind Operational Database



Schema of the Northwind Data Warehouse



Conceptual Design of the Northwind ETL: Data Sources

File Time.xls contains data for loading the Time dimension, spanning the dates in table Orders of the operational database

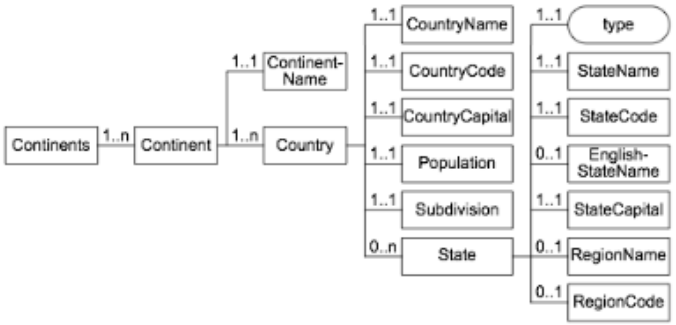
Dimensions Customer and Supplier share the geographic hierarchy starting at the City level

Data for the hierarchy State → Country → Continent loaded from Territories.xml

Start of the file Territories.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Continents>
   <Continent>
      <ContinentName>Europe</ContinentName>
      <Country>
         <CountryName>Austria</CountryName>
         <CountryCode>AT</CountryCode>
         <CountryCapital>Vienna</CountryCapital>
         <Population>8316487</Population>
         <Subdivision>Austria is divided into nine BundesInder,
            or simply Lnder (states; sing. Land).</Subdivision>
         <State type="state">
            <StateName>Burgenland</StateName>
            <StateCode>BU</StateCode>
            <StateCapital>Eisenstadt</StateCapital>
         </State>
         <State type="state">
            <StateName>Krnten</StateName>
            <StateCode>KA</StateCode>
            <EnglishStateName>Carinthia</EnglishStateName>
            <StateCapital>Klagenfurt</StateCapital>
         </State>
         ...
```

XML Schema of Territories.xml



Conceptual Design of the Northwind ETL: Data Sources

File called Cities.txt identifies to which state or province a city belongs

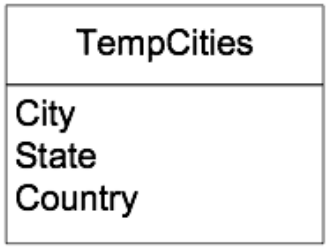
Contains three fields separated by tabs and begins as shown below

For cities located in countries that do not have states (e.g. Singapore), second field is set to null

The file is also used to identify to which state corresponds the city in the attribute TerritoryDescription of table Territories

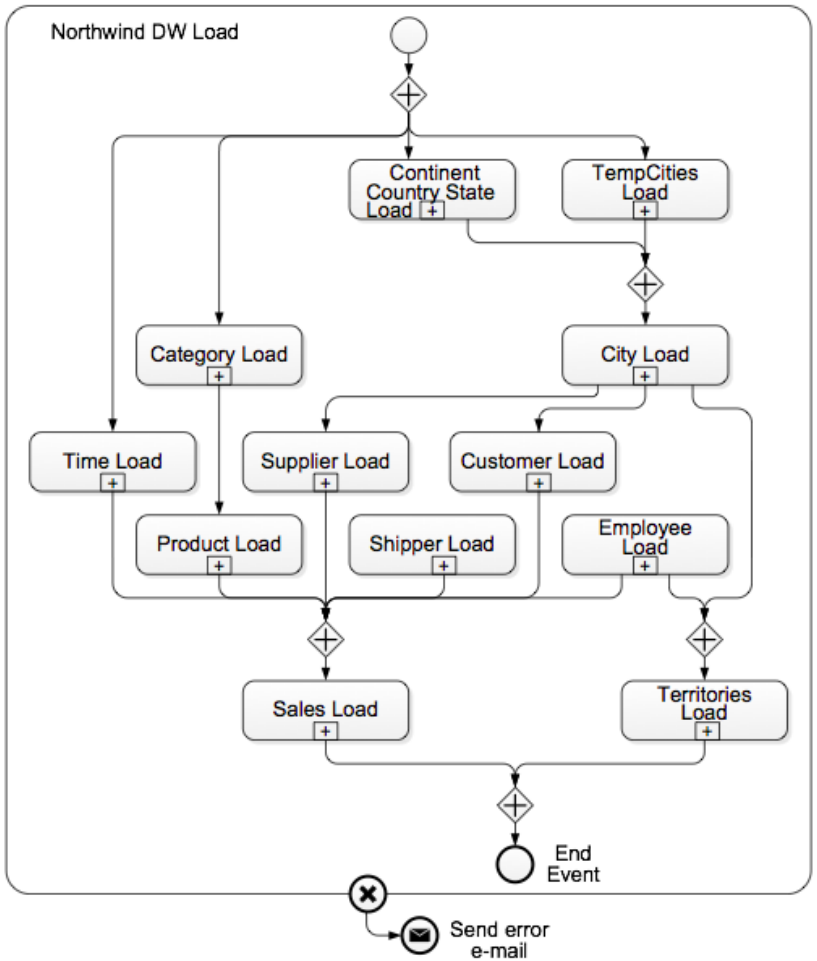
City → State → Country
Aachen → North Rhine-Westphalia → Germany
Albuquerque → New Mexico → USA
Anchorage → Alaska → USA
Ann Arbor → Michigan → USA
Annecy → Haute-Savoie → France
...

Begining of the file Cities.txt



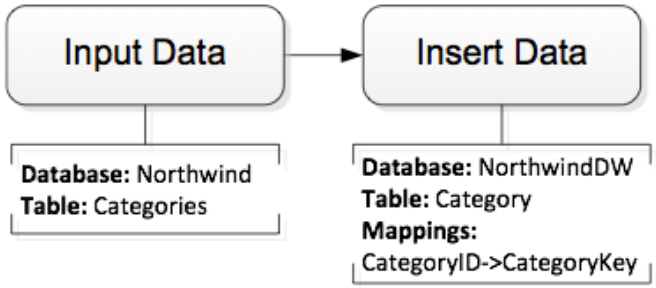
Associated table TempCities

Conceptual Design of the Northwind ETL: Overall View



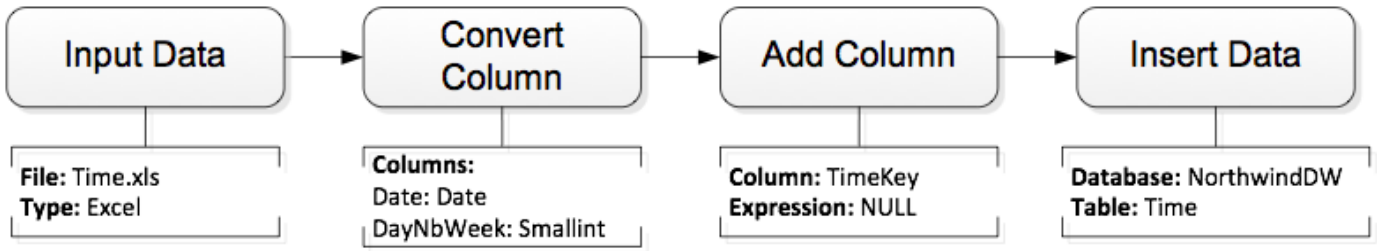
Conceptual Design of the Northwind ETL

Load of the Category dimension table



- Input task loads table Categories from the operational database
- Insert task loads the table Category in the data warehouse, mapping CategoryID to CategoryKey attribute in the Category table

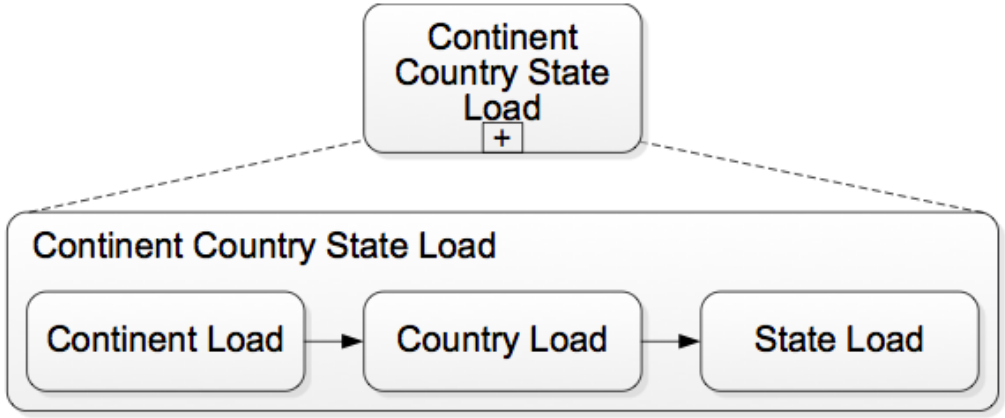
Loading the Time dimension table from an Excel file is similar, but includes a data type conversion, and an addition of the column TimeKey



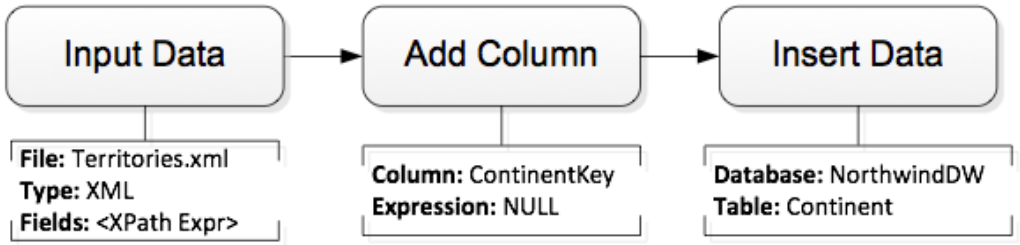
Conceptual Design of the Northwind ETL

Loading the City level first requires loading the Geography hierarchy State → Country → Continent

Associated control task



Load of the Continent table



References

A. VAISMAN, E. ZIMANYI, Data Warehouse Systems: Design and Implementation, Chapter 8 Extraction, Transformation, and Loading, Springer Verlag, 2014