Exercises Data Warehousing
ETL: Extending the Geography Dimension

We have previously created a simple SQL Server Integration Services package for the initial load of the Northwind data warehouse and then extended this package for coping with updates. We will now extend the load of the geography dimension by adding more information to it, acquired from external sources. Figure 1 shows the schema of the operational database. Furthermore, you will be using additional files with geographic information. All files can be found in the geography-data.zip on the course web page.

Your task is to rewrite the previous package and to add the richer geography dimension. It is advisable to first create a separate package for loading the geography dimension.

Before starting the assignment, please take the following steps:

1. Carefully read the documentation of the exercise.
2. Read the content of the external data files into temporary tables in your database and study their content.
3. Study for each of the tables which attributes or combinations thereof could serve as a key for the table.
4. Study which attributes the different tables have in common. Take into account that these attributes may have different names in different files. Look at the data values appearing in the columns.
5. Make a plan how to join the different tables (they do not necessarily have corresponding keys!) in order to obtain the enriched information.
6. Test if, for the attributes you selected to join on, there is referential integrity.

Note: there is some noise in the data and not for every city you will be able to find all information. It is therefore very important to obtain a good overview of the data in this exercise!

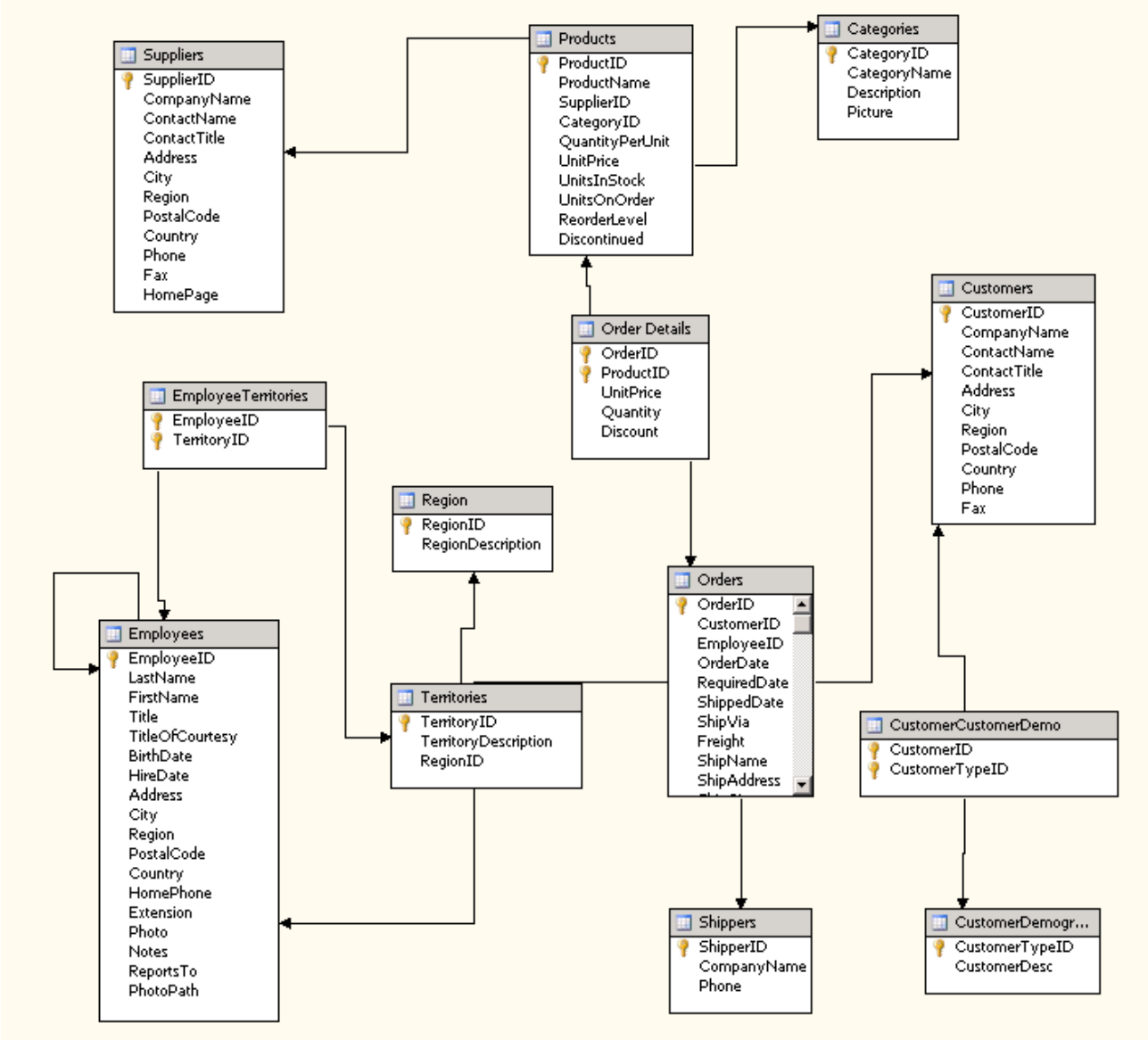
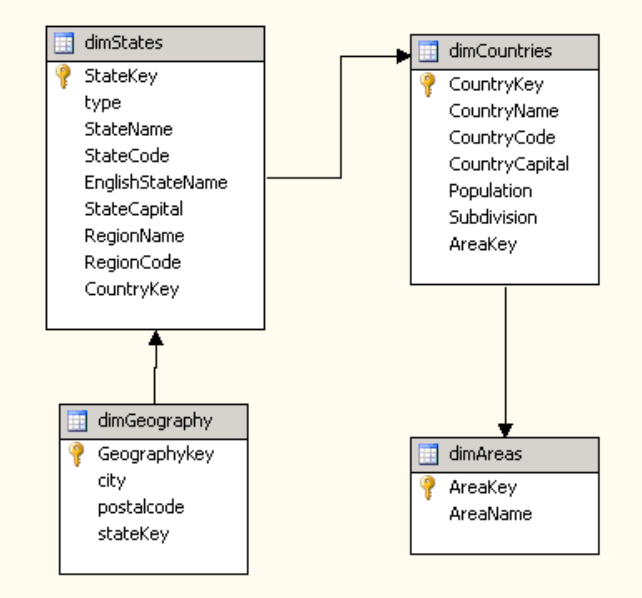


Figure 1: Schema of the Northwind operational database

Updated structure of the dimGeography dimension

Until now the geography dimension only consisted of city and country. Based on some additional files we obtained, however, we can now enrich this hierarchy. More precisely, we will add postal code and state for the cities, include detailed information about the states, add attributes for the countries such as capital city, population, and subdivision, and we will organize the countries into regions. The resulting schema in the data warehouse should look as follows:



You can find the create table statements in the file package that comes with this exercise.

Data sources for enriching the geography dimension

The data for the hierarchy `DimState`, `DimCountry`, `DimArea` is input from an XML file called `Territories.xml` that begins as follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Areas>
  <Area>
    <AreaName>Europe</AreaName>
    <Country>
      <CountryName>Austria</CountryName>
      <CountryCode>AT</CountryCode>
      <CountryCapital>Vienna</CountryCapital>
      <Population>8316487</Population>
      <Subdivision>Austria is divided into nine Bundesländer,
        or simply Länder (states; sing. Land).</Subdivision>
      <State type="state">
        <StateName>Burgenland</StateName>
        <StateCode>BU</StateCode>
        <StateCapital>Eisenstadt</StateCapital>
      </State>
      ...
```

The schema of the XML file is shown in Figure 2. Notice that `type` is an attribute of `State` and that `EnglishStateName`, `RegionName`, and `RegionCode` are optional.

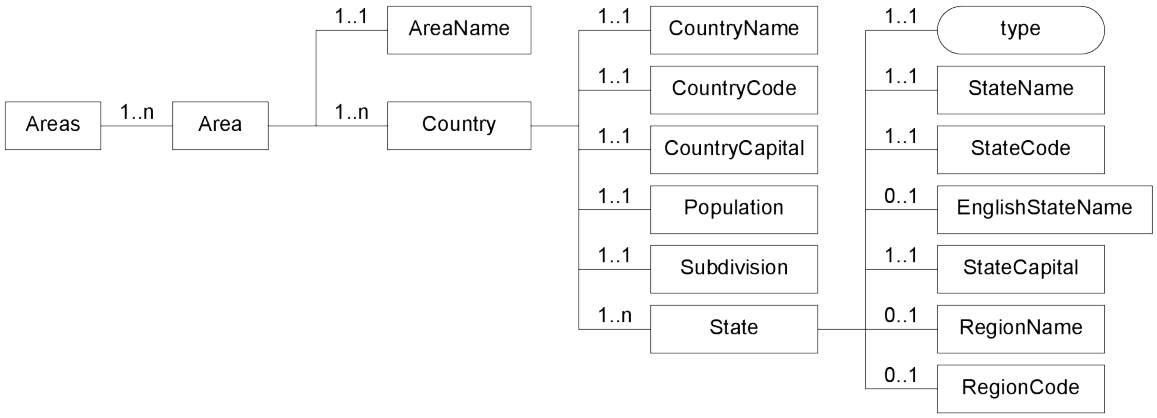


Figure 2: XML Schema of the file `Territories.xml`

In SSIS, you can load XML files using the "XML source", available in the Data Flow Sources category of the Data Flow tab. Use the xml schema when loading the file.

The `DimGeography` dimension is obtained from the `City`, `Region`, `PostalCode`, and `Country` from both `Customers` and `Suppliers`. Notice that the attribute `Region` contains in fact a state name (e.g., `Québec`) or a state code (e.g., `CA`); similarly, the attribute `Country` contains a country name (e.g., `Canada`) or a country code (e.g., `USA`). To identify to which state corresponds a city the file `cities.txt` is used. The file contains three fields separated by tabs and begins as follows:

```
Atlanta → Georgia → USA
Austin → Texas → USA
Beachwood → Ohio → USA
Bedford → Indiana → USA
...
```

This file is also used to identify to which state correspond the attribute `TerritoryDescription` in `DimTerritory`, which in fact contains city names from the United States.

If you have successfully finished this and previous exercises, create a new SSAS project to construct a cube on the basis of this data warehouse.