

**ASSIGNMENT 2 FOR THE COURSE
DATA MANAGEMENT, BUSINESS INTELLIGENCE AND
VISUALIZATION**

**“CREATE A STAR SCHEMA DATATABASE WITH ETL
PROCCES IN SQL SERVER AND VISUAL STUDIO”**

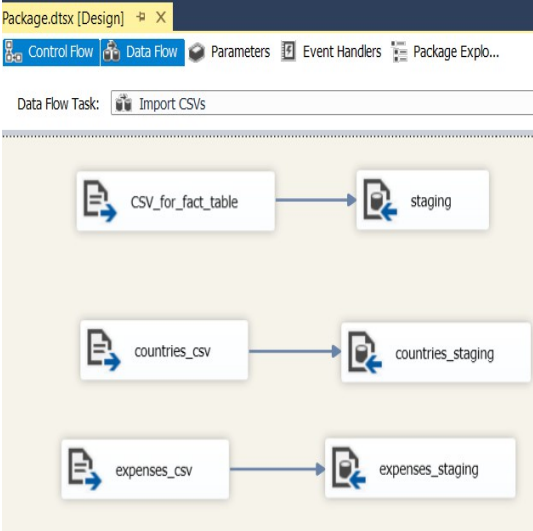
Panagiotis Petrocheilos

For the second assignment of the program I created a star schema database with costs for different goods /services and salaries around the world, expressed in dollars. My target is to compare the data in city, country, sub region and continent level. I used the dataset cost-of-living from kaggle ([Global Cost of Living \(kaggle.com\)](https://www.kaggle.com/datasets/numbeo/cost-of-living)). The data are extracted from the site www.numbeo.com which is a world statistic database.

Because the dataset mentioned only cities and countries, I also used a csv file with the countries divided by intermediate region (e.g. Balkans) , sub region (e.g. South Africa) and Continent. Finally I created a third excel to divide the types of expenses by category (e.g. food/drink, clothes) and priority (Basic goods are first, consumer goods are second and luxury goods are third).

Creating the SSIS project

The first step was to import the three csv files to the database. I chose unicode string data type for the string columns (dt_wstr) because I found it more convenient than the dt_str type which was suggested by the program. Through the connection wizard, I created the three equivalent tables in the Sql server management studio. I also created an sql task to truncate the three tables before the import



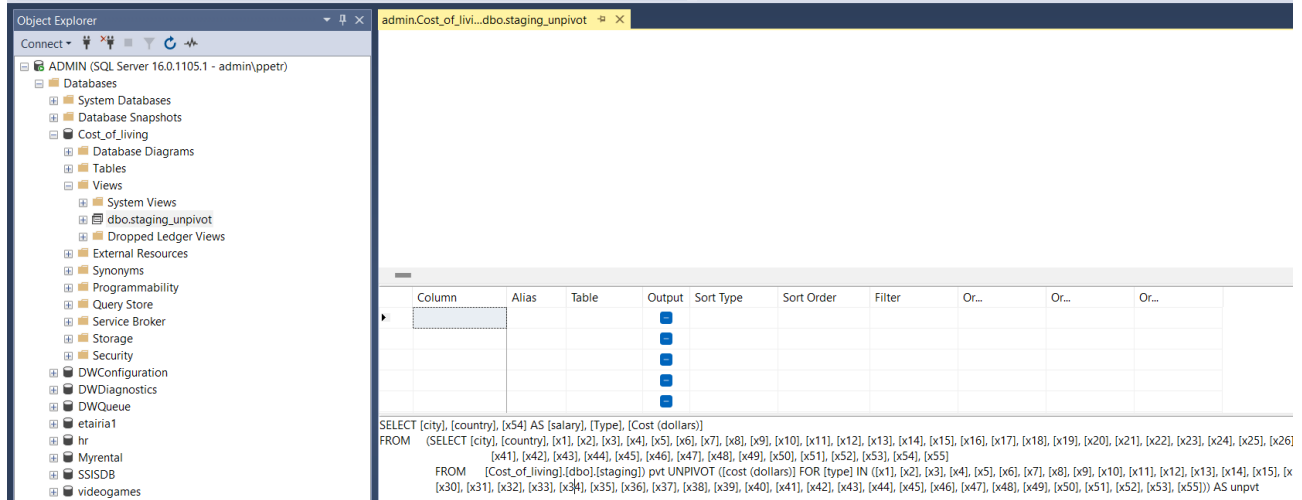
Column Name	Data Type	Allow Nulls
city	nvarchar(50)	<input checked="" type="checkbox"/>
country	nvarchar(50)	<input checked="" type="checkbox"/>
x1	float	<input checked="" type="checkbox"/>
x2	float	<input checked="" type="checkbox"/>
x3	float	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
[sub-region]	nvarchar(50)	<input checked="" type="checkbox"/>
[intermediate-region]	nvarchar(50)	<input checked="" type="checkbox"/>
region	nvarchar(50)	<input checked="" type="checkbox"/>
Country_label	nvarchar(50)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
code	nvarchar(50)	<input checked="" type="checkbox"/>
description	nvarchar(300)	<input checked="" type="checkbox"/>
Category	nvarchar(50)	<input checked="" type="checkbox"/>
priority	nvarchar(50)	<input checked="" type="checkbox"/>

Unpivoting the staging table

In the staging table , each expense had its own column. This was not convenient for the analysis, so I created a view in SSMS which had only one column with the type of expense and a second one for its cost.



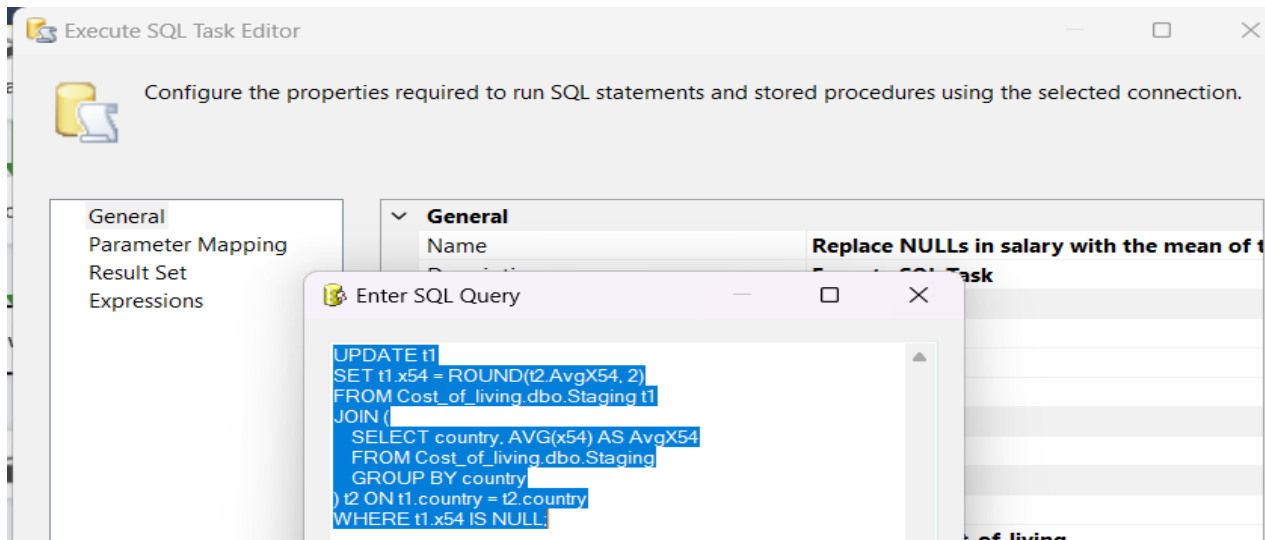
```

SELECT [city], [country], [x54] AS [salary], [Type], [Cost (dollars)]
FROM (SELECT [city], [country], [x1], [x2], [x3], [x4], [x5], [x6], [x7], [x8], [x9], [x10], [x11], [x12], [x13], [x14], [x15], [x16], [x17], [x18], [x19], [x20], [x21], [x22], [x23], [x24], [x25], [x26], [x41], [x42], [x43], [x44], [x45], [x46], [x47], [x48], [x49], [x50], [x51], [x52], [x53], [x54], [x55]
FROM [Cost_of_Living].[dbo].[staging]) pvt UNPIVOT ([cost (dollars)] FOR [type] IN ([x1], [x2], [x3], [x4], [x5], [x6], [x7], [x8], [x9], [x10], [x11], [x12], [x13], [x14], [x15], [x30], [x31], [x32], [x33], [x34], [x35], [x36], [x37], [x38], [x39], [x40], [x41], [x42], [x43], [x44], [x45], [x46], [x47], [x48], [x49], [x50], [x51], [x52], [x53], [x55])) AS unpvt
  
```

Handling the NULL values

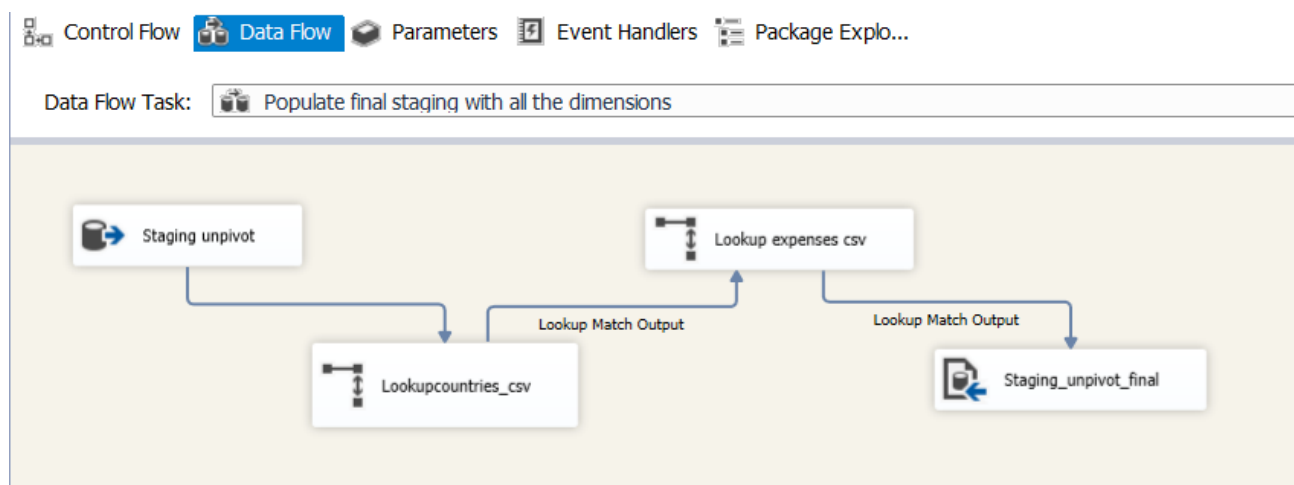
In the dataset there are some NULL values (They are called nan but with ignore failure in the import task they pass as NULL in float data type columns). For now I did not make any transformation to most of them because I wait to see if they will create problem to the analysis (It is ok if some costs will have comparison between lesser number of countries). The only value I handled was the salary column, because salary is an important metric of the dataset for comparison and I am thinking to use percentages between the various costs and the salaries.

For the NULL values in this column I created an sql task after the import. It replaces the NULL value of the city salary with the average salary of the specific country.



Create the final staging table with all the facts

Because until now I had three staging tables, I decided to make the simplest star schema I could. With a dataflow task, I created two look up transformations to the staging table, first adding the extra columns from the countries staging table and then doing the same for the expenses staging table. I gathered all the columns in one table to be ready to create the fact table with all the data.



Creating and updating the dimensions

In the next step I created all the dimension tables in SSMS with automatic identity for every new ID. I created an index to allow only unique values for the labels and ignore duplicate entries. Finally I created an sql task in SSIS to update every dimension with the data from the final staging table.

The screenshot displays the 'Indexes/Keys' dialog box in SQL Server Enterprise Manager. The 'Selected Primary/Unique Key or Index' list contains 'IX_city_dim' and 'PK_city_dim'. The 'Editing properties for existing primary/unique key or index' section shows the following details:

- General**
 - Columns: city_label (ASC)
 - Is Unique: Yes
 - Type: Index
- Identity**
 - (Name): IX_city_dim
 - Description: IX_city_dim
- Table Designer**
 - Create As Clustered: No
 - Data Space Specification: PRIMARY
 - Fill Specification: PRIMARY
 - Ignore Duplicate Keys: Yes

The background shows the 'Column Properties' window for the 'city_dim' table, with columns 'city_id' (int, identity) and 'city_label' (nvarchar(50)).

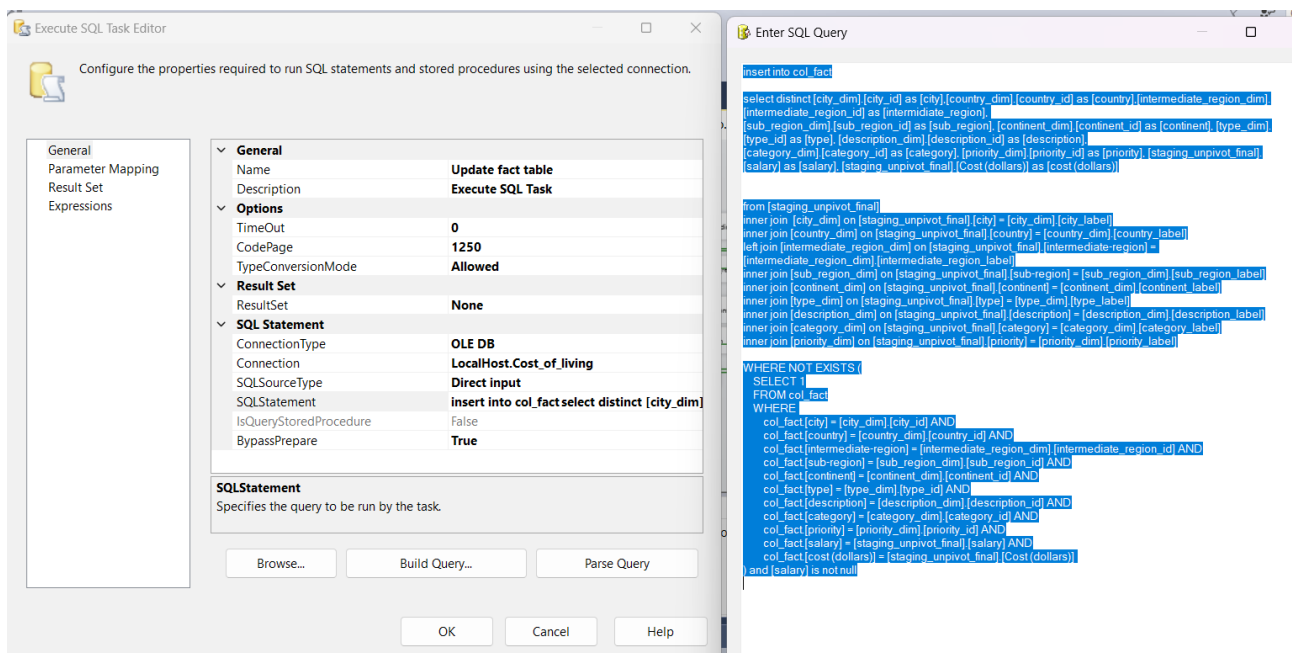
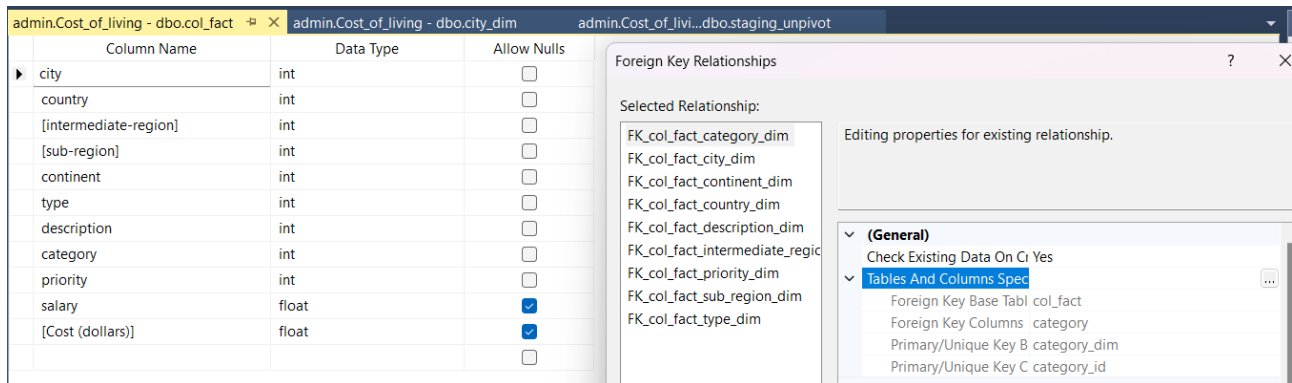
The screenshot shows the 'Execute SQL Task Editor' window. The 'General' tab is selected, displaying the task name 'update city_dim' and the description 'Execute SQL Task'. The 'SQL Statement' tab is active, showing the SQL query:

```
insert into city_dim(city_label) select distinct [city] from [Cost_of_living].[dbo].[staging_unpivot]
```

The 'Enter SQL Query' dialog box is open, showing the query text.

Creating and updating the fact table

In the last step, I created the fact table with connection to all the dimensions , having salary and cost as metrics in SSMS. I created the foreign key relationships with the dimensions. Finally I created an sql task in SSIS to update it. In the sql task I put a condition for checking if the row already exists so not to create duplicate rows if the table is updated regularly (Also to check if the salary is NULL because this is the only column with NULL values and these rows are not recognized as duplicate even if they already exist).



Final comments

The fact table has approximately 230.000 rows. Bellow there are two screenshots of the project in its final form. The first is the SSIS project control flow and the second is the star schema depicted in SSAS

