UNIVERSITÀ DI PISA

Report Project Group 13

Laboratory of Data Science
AY 2022/2023

Cosimo Faeti 636812
Riccardo Galarducci 637763

Part 1

In this part of the project assignment, we created a data mart following the schema assigned. Then we extracted data from the file *answerdatacorrect.csv* and *subject_metadata.csv* and transformed them according to the pipeline in Assignment 1. Finally, we populate the data mart using the processed data.

Assignment 0

In this assignment the goal was to create a data mart according to the given schema using SQL Server Management Studio.
In the following are listed the tables of the data mart and the corresponding attributes and data type:

- **Answers**: *answerid* (int), *questionid* (int), *userid* (int), *organizationid* (int), *dateid* (int), *subjectid* (nvarchar(50)), *answer_value* (int), *correct_answer* (int), *iscorrect* (int), *confidence* (int)
- **Date**: *dateid* (int), *date* (nvarchar(50)), *day* (nvarchar(50)), *month* (nvarchar(50)), *year* (int), *quarter* (nvarchar(50))
- **Geography**: *geoid* (int), *region* (nvarchar(max)), *country_name* (nvarchar(max)), *continent* (nvarchar(50))
- **Organization**: *organizationid* (int), *groupid* (int), *quizid* (int), *schemeofworkid* (int)
- **Subject**: *subjectid* (nvarchar(50)), *description* (nvarchar(max))
- **Users**: *userid* (int), *dateid* (int), *geoid* (int), *gender* (int)

A particular attention was placed on respecting the relationships between the tables, especially with regard to the constraints between the fact table (*Answers*) and the dimension tables by establishing a relation between primary and foreign key according to the schema (Figure 1).

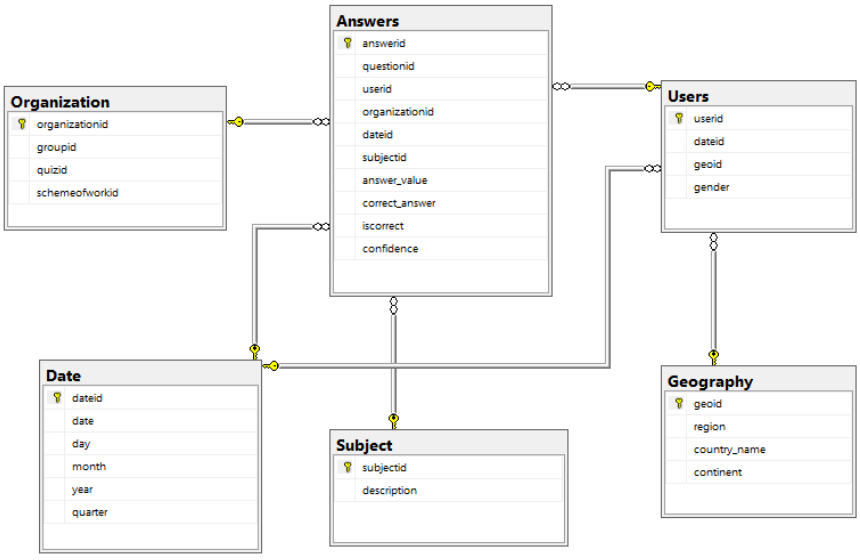

Figure 1: Datawarehouse schema

Assignment 1

The general idea was to develop a Python program that was maintainable, flexible and reusable. In particular, we used a dictionary as main data structure to model the structure of the data mart tables in such a way as to include natural key/surrogate key as key in order to easily retrieve the associated values in constant time (*O(1)*). To each key is associated a list which stores the attribute values.
In the following, we reported a brief description for each tables:

- **Answers**: Answers is the fact table that counts 538.835 rows and as natural key has *answerid*. Each rows in the table is identified by distinct values of the composition of the foreign keys (*userid*, *organizationid*, *dateid* and *subjectid*). The main measure of the data warehouse is *iscorrect* that is computed by comparing the variables *answer_value* and *correct_answer*, while other attributes are *questionid* and *confidence*

- **Date**: Date table counts 596 rows and as surrogate key has *dateid*. It is composed by the distinct date retrieved from *DateOfBirth* and *DateAnswered* columns. In particular, we have performed an additional cleaning operation on *DateAnswered* to discard hours and minutes. The attributes are *date*, *day*, *month*, *quarter* and *year*. The value of the dimension attributes have been implemented in order to establish a hierarchy between them to enable future analysis

- **Geography**: Geography table counts 76 rows and as surrogate key has *geoid*. The attributes are *region*, *country_name* and *continent* where the last two has been defined exploiting the country code through external libraries

- **Organization**: Organization table counts 24.640 rows and as surrogate key has *organizationid*. It is composed by the distinct combination of the attributes *groupid*, *quizid* and *schemeofworkid*

- **Subject**: Subject table counts 412 rows and as natural key has *subjectid*, which consists of distinct list of indexes corresponding to subjects in *subject_metadata.csv*. The only attribute is *description* that is a string composed of topics retrieved from the *subject_metadata.csv* and ordered on the level attribute

- **Users**: Users table counts 13.630 rows and as surrogate key has *userid*, two foreign keys (*dateid* and *geoid*) and an additional dimensional attribute *gender*. *dateid* is foreign key of Date table and refers to the date of birth of the user, while *geoid* is foreign key that model the relationship with the Geography table that is the id of the location of birth

Assignment 2

As final step, each dictionary is converted into a csv file and in order to populate the database it has been followed the connection protocol exploiting the *pyodbc* library.

Part 2

In this part of the project assignment, we solved three assignments by using Sequel Server Integration Services (SSIS) with computation only on client side. The SSIS project is composed by three SSIS packages that corresponds to each assignment.
We used the database created in the previous part.

Assignment 0

The statement for the Assignment 0 is: *For every subject, the number of correct answers of male and female students.*

The complete design is shown in Figure 2.

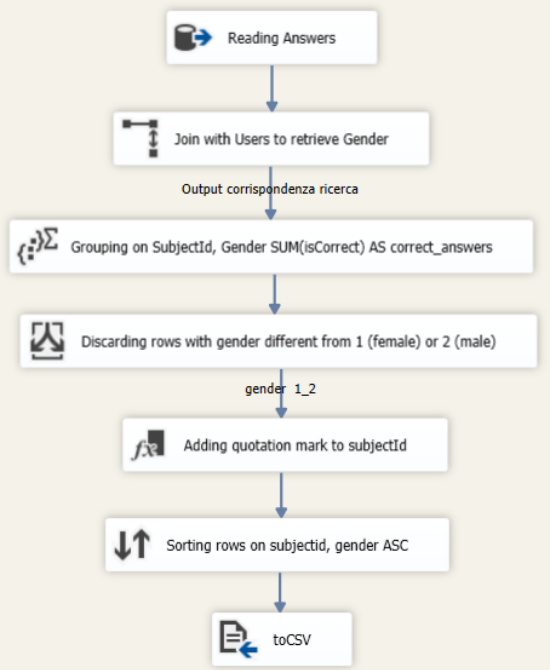

Figure 2: Data Flow for Assignment 0

The first data flow task is composed by the nodes which are listed as follows:

1. **OLE DB Source**. Specifying an OLE DB connection with the database *Group_13* and accessing the *Answers* table importing the following columns: *answerid*, *subjectid*, *iscorrect* and *userid*;
2. **Lookup**. Retrieving *gender* by joining *Answers* with *Users* table on *userid*;
3. **Aggregate**. Performing a grouping on *subjectid* and *gender*, and computing the sum over *iscorrect* labeled as *correct_answers*;
4. **Conditional split**. Discarding rows with values other than 1 and 2 in the *gender* column by using the condition: "[gender] == 1 || [gender] == 2";
5. **Derived column**. Adding quotation mark to *subjectid* in order to avoid problems when exporting to .csv file;
6. **Sort**. Sorting rows on *subjectid* and *gender* in ascending order;
7. **File flat destination**. Exporting the result on a .csv file.

Assignment 1

The statement for the Assignment 1 is: *A subject is said to be easy if it has more than 90% correct answers, while it is said to be hard if it has less than 20% correct answers. List every easy and hard subject, considering only subjects with more than 10 total answers.*

The complete design is shown in Figure 3.

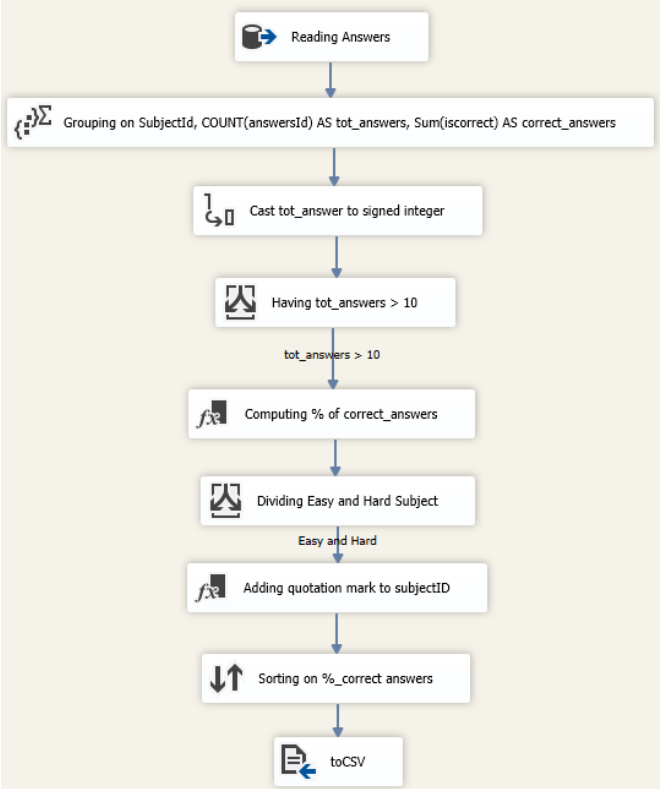

Figure 3: Data Flow for Assignment 1

The second data flow task is composed by the nodes which are listed as follows:

1. **OLE DB Source**. Specifying an OLE DB connection with the database *Group_13* and accessing the *Answers* table importing the following columns: *answerid*, *subjectid* and *iscorrect*;

2. **Aggregate**. Performing a grouping on *subjectid*, computing the sum over *iscorrect* labeled as *correct_answers* and counting over *answerid* labeled as *tot_answers*;

3. **Data Conversion**. Converting *tot_answers* from an eight-byte unsigned integer to eight-byte signed integer, otherwise we could not perform the conditional split;

4. **Conditional split**. Discarding rows with values of *tot_answers* greater than 10;

5. **Derived column**. Computing the percentage of *correct_answers* (labeled as *%_correct_answers*): $100 \cdot \frac{\text{correct\_answers}}{\text{tot\_answers}}$;

6. **Conditional split**. Filtering *easy* and *hard* subject with percentage of *correct_answers* $<$ 20 and $>$ 90 respectively, by using the condition: "[%_correct_answers]<20 || [%_correct_answers]>90";

7. **Derived column**. Adding quotation mark to *subjectid* in order to avoid problems when exporting to .csv file;

8. **Sort**. Ordering the percentage of *correct_answers* in descending order and passing through only *subjectid* and *%_correct_answers*;

9. **Flat file destination**. Exporting the result on a .csv file.

Assignment 2

The statement for the Assignment 2 is: *For each country, the student or students that answered the most questions correctly for that country.*

The complete design is shown in Figure 4.

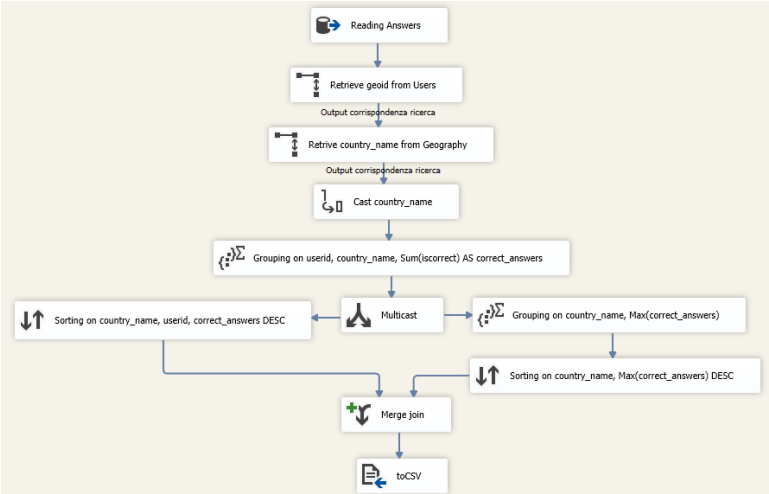

Figure 4: Data Flow for Assignment 2

The third data flow task is composed by the nodes which are listed as follows:

1. **OLE DB Source**. Specifying an OLE DB connection with the database *Group_13* and accessing the *Answers* table importing the following columns: *answerid*, *userid* and *iscorrect*;
2. **Aggregate**. Performing a grouping on *userid* and computing the sum over *iscorrect* labeled as *correct_answers*;
3. **Lookup**. Retrieving *geoid* by joining *Answers* with *Users* table on *userid*;
4. **Lookup**. Retrieving *country_name* by joining *Answers* with *Geography* table on *geoid*;
5. **Data Conversion**. Converting *country_name* data type from a Unicode character string to null-terminated Unicode character string in order to perform the grouping on *country_name*;
6. **Aggregate**. Performing a grouping on *userid* and casted *country_name*, and computing the sum over *iscorrect* labeled as *correct_answers*;
7. **Multicast**. Distributing the input to multiple outputs;
8. **Sort**. On the left side, ordering the *correct_answers* in descending order, while converted *country_name* and *userid* in ascending order;
9. **Grouping**. On the right side, performing a grouping on converted *country_name* and computing the max over *correct_answers* labeled as *max_correct_answers*;
10. **Sort**. Ordering *Max_correct_answers* in descending order with option of removing rows with duplicate sort values, while converted *country_name* in ascending order;
11. **Merge join**. Merging the two data flows (8) and (10) which are sorted on the values of *correct_answers* and converted *country_name*;
12. **Flat file destination**. Exporting the result on a .csv file.

Part 3

In this part of the project assignment, we first built a datacube by using SQL Server Analysis Services (SSAS) from the data stored in the database (Assignment 0). Then, we solved the query of Assignment 1, 2 and 3 using MDX language in Microsoft SQL Server Management. AS last step, we created two dashboards using PowerBi (Assignment 4 and 5).

Assignment 0

As first step, we create a data source based on a connection with the database *Group_13_DB* and we defined a data source view selecting all the tables.
Following we defined the multilevel hierarchies by creating new dimensions for Geography and Date, as *RegionCountryContinent* and *DayMonthQuarterYear* respectively, maintaining the existing flat hierarchies.
The definition of the cube is performed by selecting the fact table of the star schema, named *Answers*, which contains the following measures:

- *NAnswers*, which is the number of answers
- *CorrectAnswers*, which is *iscorrect*
- *IncorrectAnswers*, which is the computed measure using the following SQL expression: CASE iscorrect WHEN 1 THEN 0 ELSE 1 END

There is an additional measure named *NUsers* contained in *Users* table.

Assignment 1

The statement of the query is: *Show the student that made the most mistakes for each country.*

```
select [Measures].[IncorrectAnswers] on columns,
       generate(
                [Geography].[Country Name].[Country Name],
       topcount( ([Geography].[Country Name].currentmember,
                  [Users].[Userid].[Userid]), 1, [Measures].[IncorrectAnswers] )
       ) on rows
from [Group 13 DB]
```

Figure 5: Assignment 1

In particular, the *generate()* function takes the *Country Name* as first set and *TopCount* function as second set. The latter returns the user with the highest number of incorrect answers for the current member.

Assignment 2

The statement of the query is: *For each subject, show the student with the highest total correct answers.*

```
select [Measures].[CorrectAnswers] on columns,
       nonempty(generate(
                [Subject].[Subjectid].[Subjectid],
       topcount( ([Subject].[Subjectid].currentmember,
                  [Users].[Userid].[Userid]), 1, [Measures].[CorrectAnswers] )
       )) on rows
from [Group 13 DB]
```

Figure 6: Assignment 2

In particular, the *generate()* function takes the *Subjectid* as first set and *TopCount* function as second set. The latter returns the user with the highest number of correct answers for the current member. In addition, it has been added *NonEmpty()* function to avoid null values.

Assignment 3

The statement of the query is: *For each continent, show the student with the highest ratio between his total correct answers and the average correct answers of that continent.* In particular,

```
with member AvgCorrectAnswers as
        avg(
            ([Geography].[Continent],
            [Users].[Userid].[Userid]),
            [Measures].[CorrectAnswers]
            )
member Ratio as
            [Measures].[CorrectAnswers] / AvgCorrectAnswers ,
format_string = "percent"

select {[Measures].[CorrectAnswers],AvgCorrectAnswers, Ratio} on columns,
       generate(
                [Geography].[Continent].[Continent],
       topcount(([Geography].[Continent].currentmember,
                  [Users].[Userid].[Userid]), 1, Ratio )
                ) on rows
from [Group 13 DB]
```

Figure 7: Assignment 3

we defined two calculated members, *AvgCorrectAnswers* and *Ratio* respectively. The former retrieve the average correct answers for each continent, while the latter computes the ratio between the total correct answers and *AvgCorrectAnswers*. The main of the query exploits the *generate()* function that takes the *Continent* as first set and *TopCount* function as second set. The latter returns the user with the highest ratio for the current member.

Assignment 4

We created a dashboard that shows the geographical distribution of correct and incorrect answers. In particular, the dashboard is composed by five charts:

- a *clustered column chart*, which represents on the x-axis the geographical hierarchy and on y-axis the number of correct answers, incorrect answers and number of total answers
- two *pie chart*, one for each measures by the geographical hierarchy
- two *map*, one for each measures by the geographical hierarchy

Assignment 5

We created a dashboard that shows the temporal distribution of correct and incorrect answers. In particular, the dashboard is composed by four charts:

- a *stacked area chart*, which represents on the x-axis the temporal hierarchy and on y-axis the number of correct answers, incorrect answers and number of total answers
- a *stacked column chart*, which represents on the x-axis the temporal hierarchy and on y-axis the percentage of correct and incorrect answers
- two *map*, one for each measures by geographical hierarchy and Year