Data Warehousing, MIRI, ESSI, UPC

Multidimensional design

Training session

2018

This training session is focused on the requirement-driven design of multidimensional schema (MD) in a data warehouse (DW).

Prerequisites:

- All examples are created over the LEARN-SQL system. For better understanding of the system under the study, the schematic/diagrammatic representation of the ontology is showed in Figure 1.
- During the training session as well as in the lab, we will use the *Indyco Builder* tool for conceptual modeling of a DW.
    - Conceptual modeling of a MD schema
    - Uses Dimensional Fact Model (DFM)
    - Intuitive graphical user interface
    - Validation
- We strongly suggest you to check more details and get familiar with the *Indyco Builder* tool following the given links.
    - Dimensional Fact Model (DFM)
        - First publication: (Golfarelli, Maio, & Rizzi, 1998)
        - Wiki: http://en.wikipedia.org/wiki/Dimensional_Fact_Model
        - Extended guidelines: http://www.indyco.com/kb/dimensional-fact-model/
    - Indyco Builder tool
        - Official web page: http://www.indyco.com/builder/
        - User manual: https://www.youtube.com/watch?v=haXLU0M2PZQ
        - Download: http://www.indyco.com/start-here/request-an-educational-license/ (follow the instructions in Learn SQL on how to download and install Indyco Builder).

This training session has two assignments:

Assignment 1:
Requirement-driven design of a multidimensional schema

Considering the diagrammatic (graphical) ontology representation of the Learn-SQL domain in Figure 1, determine the multidimensional context of the given requirement inside the given ontology and produce valid MD schema solution(s) that satisfy the requirement at hand.

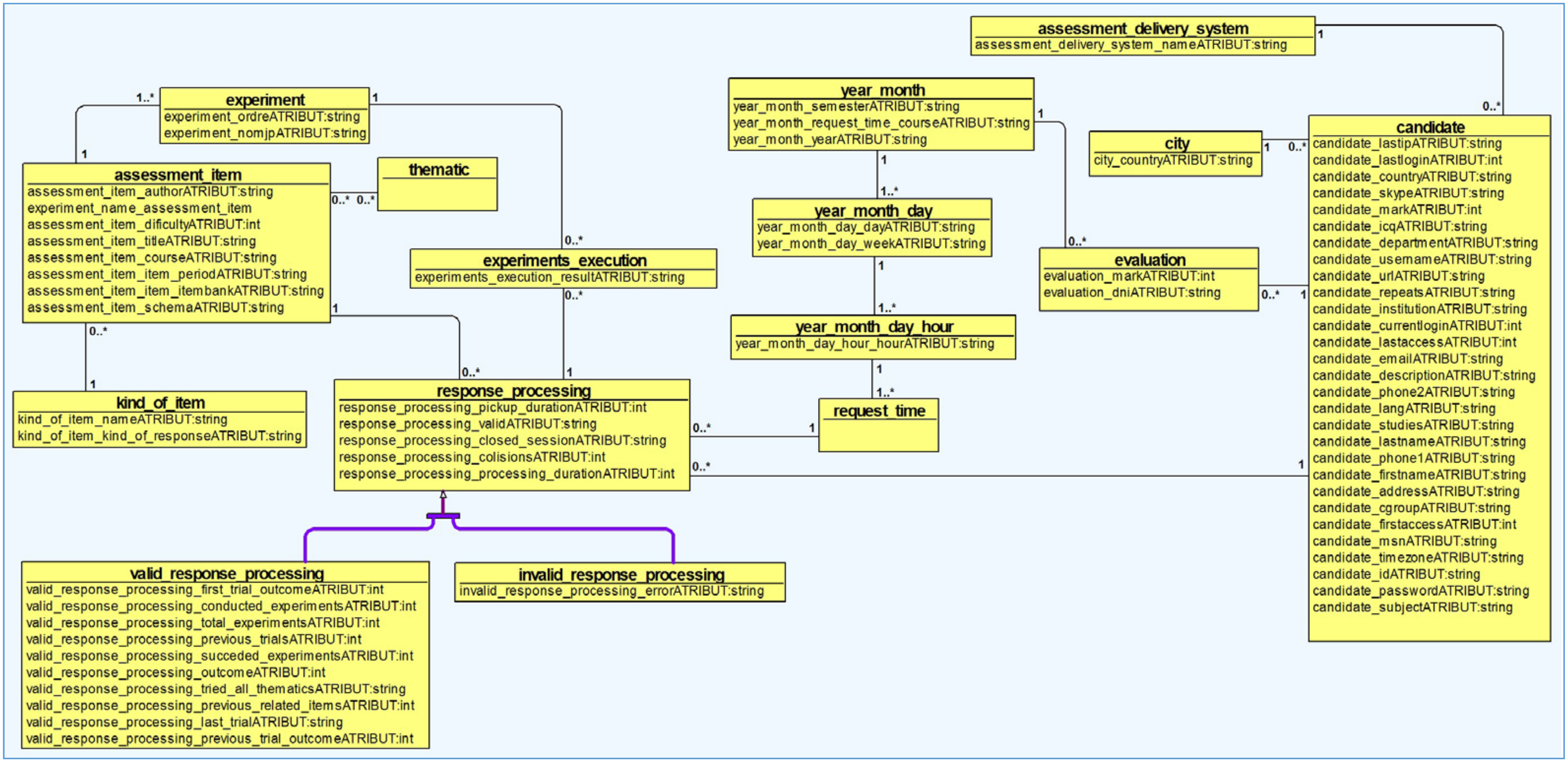

assessment_delivery_system
assessment_delivery_system_nameATRIBUT:string
experiment
experiment_ordreATRIBUT:string
experiment_nomjpATRIBUT:string
year_month
year_month_semesterATRIBUT:string
year_month_request_time_courseATRIBUT:string
year_month_yearATRIBUT:string
city
city_countryATRIBUT:string
candidate
candidate_lastipATRIBUT:string
candidate_lastloginATRIBUT:int
candidate_countryATRIBUT:string
candidate_skypeATRIBUT:string
candidate_markATRIBUT:int
candidate_icqATRIBUT:string
candidate_departmentATRIBUT:string
candidate_usernameATRIBUT:string
candidate_urlATRIBUT:string
candidate_repeatsATRIBUT:string
candidate_institutionATRIBUT:string
candidate_currentloginATRIBUT:int
candidate_lastaccessATRIBUT:int
candidate_emailATRIBUT:string
candidate_descriptionATRIBUT:string
candidate_phone2ATRIBUT:string
candidate_langATRIBUT:string
candidate_studiesATRIBUT:string
candidate_lastnameATRIBUT:string
candidate_phone1ATRIBUT:string
candidate_firstnameATRIBUT:string
candidate_addressATRIBUT:string
candidate_cgroupATRIBUT:string
candidate_firstaccessATRIBUT:int
candidate_msnATRIBUT:string
candidate_timezoneATRIBUT:string
candidate_idATRIBUT:string
candidate_passwordATRIBUT:string
candidate_subjectATRIBUT:string
assessment_item
assessment_item_authorATRIBUT:string
experiment_name_assessment_item
assessment_item_dificultyATRIBUT:int
assessment_item_titleATRIBUT:string
assessment_item_courseATRIBUT:string
assessment_item_item_periodATRIBUT:string
assessment_item_item_itembankATRIBUT:string
assessment_item_schemaATRIBUT:string
thematic
year_month_day
year_month_day_dayATRIBUT:string
year_month_day_weekATRIBUT:string
evaluation
evaluation_markATRIBUT:int
evaluation_dniATRIBUT:string
experiments_execution
experiments_execution_resultATRIBUT:string
year_month_day_hour
year_month_day_hour_hourATRIBUT:string
kind_of_item
kind_of_item_nameATRIBUT:string
kind_of_item_kind_of_responseATRIBUT:string
response_processing
response_processing_pickup_durationATRIBUT:int
response_processing_validATRIBUT:string
response_processing_closed_sessionATRIBUT:string
response_processing_colisionsATRIBUT:int
response_processing_processing_durationATRIBUT:int
request_time
valid_response_processing
valid_response_processing_first_trial_outcomeATRIBUT:int
valid_response_processing_conducted_experimentsATRIBUT:int
valid_response_processing_total_experimentsATRIBUT:int
valid_response_processing_previous_trialsATRIBUT:int
valid_response_processing_succeded_experimentsATRIBUT:int
valid_response_processing_outcomeATRIBUT:int
valid_response_processing_tried_all_thematicsATRIBUT:string
valid_response_processing_previous_related_itemsATRIBUT:int
valid_response_processing_last_trialATRIBUT:string
valid_response_processing_previous_trial_outcomeATRIBUT:int
invalid_response_processing
invalid_response_processing_errorATRIBUT:string
1
0..*
1..*
1
1
1
1..*
1
0..*
1
0..* 0..*
0..*
1
1..*
0..*
0..*
1
1
0..*
1..*
1
0..*
1
1
0..*
1
1..*
0..*
1
0..*
1

Figure 1: Diagrammatic representation of the domain LEARN-SQL ontology

Assignment 1

Requirement-driven design of a multidimensional schema

New information requirement is posed by the Learn-SQL user:

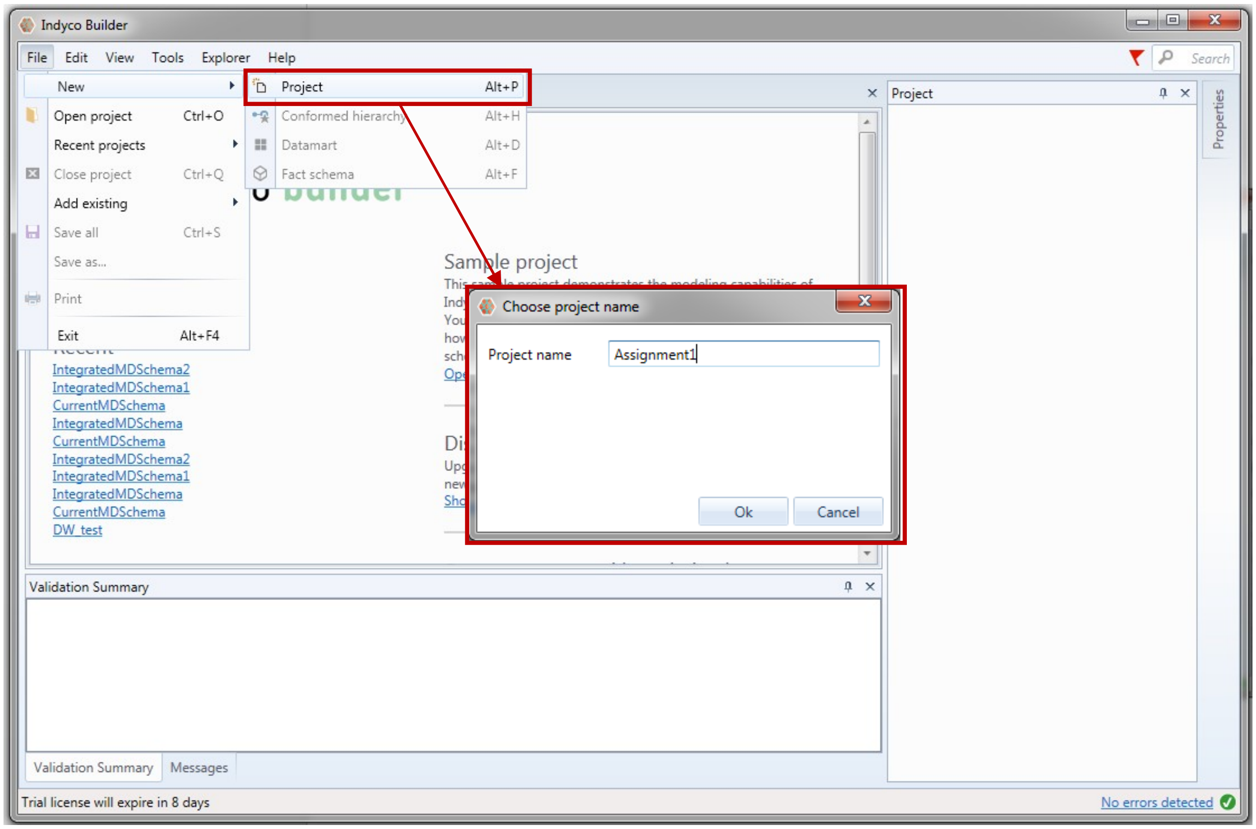
“I want to analyze the average processing duration for each week and candidate where the difficulty level of the assignment item is higher than 7.”

Considering the diagrammatic (graphical) ontology representation of the Learn-SQL domain in Figure 1, determine the multidimensional context of the given requirement inside the given ontology and produce valid MD schema solution(s) that satisfy the requirement at hand.

Important note: Different alternative solutions (if more than one) should be provided in different data marts inside the same Indyco project.

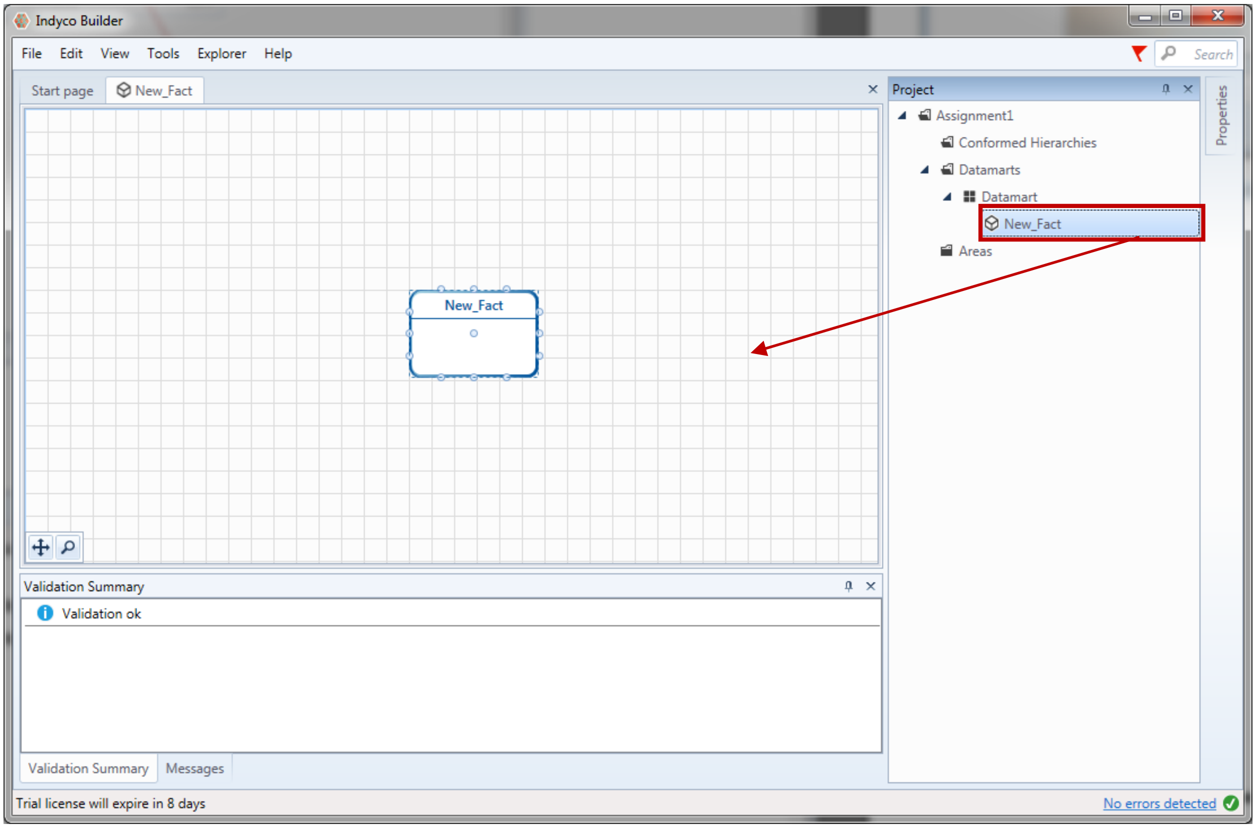
MD schema design

1. **Opening the Indyco Builder tool.** After installing Indyco Builder, you should open it and create new project, as showed in the figure below



Project name: Assignment1

After we created a new project, the empty project with an initial empty fact will appear on the right as showed in the figure below. After double-click on the fact, the canvas where the MD design is created for that fact should be visible in the screen as shown in the figure below.



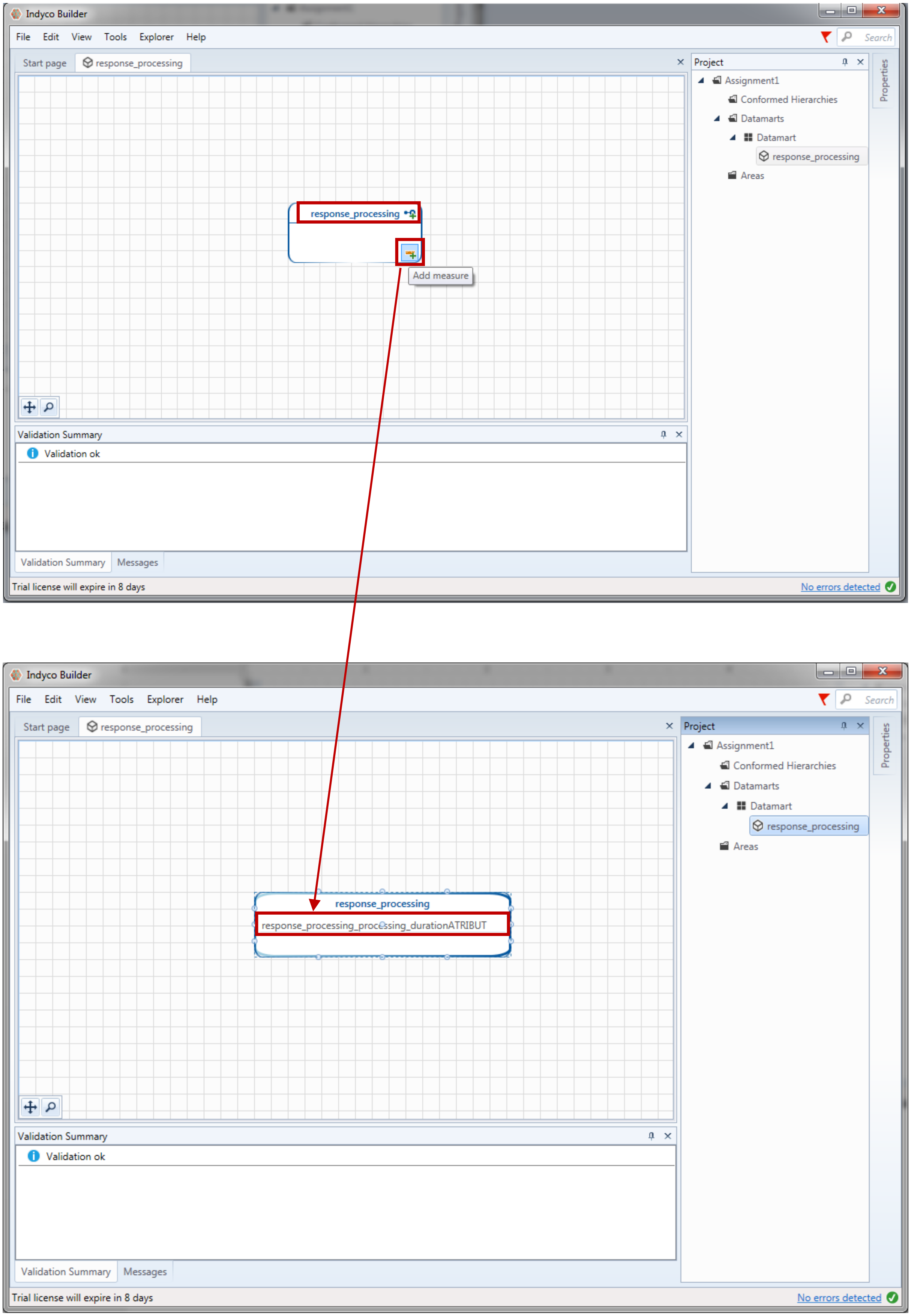
Validation Summary

Validation ok

Trial license will expire in 8 days

No errors detected

2. **Define the focus of the analysis.** Starting from the above business requirement, we should first identify how the focus of the analysis (a potential fact) maps over the domain ontology (Figure 1). Thus, in our example, we identify that "average processing duration" can map to *response_processing* concept in the LEARN-SQL ontology, more precisely to the attribute *response_processing_processing_durationATRIBUT*. Thus we insert a new (or modify the initial empty) fact, and further add its measure (i.e., *processing_duration*), as showed in the figures below.



Validation ok

Trial license will expire in 8 days

No errors detected

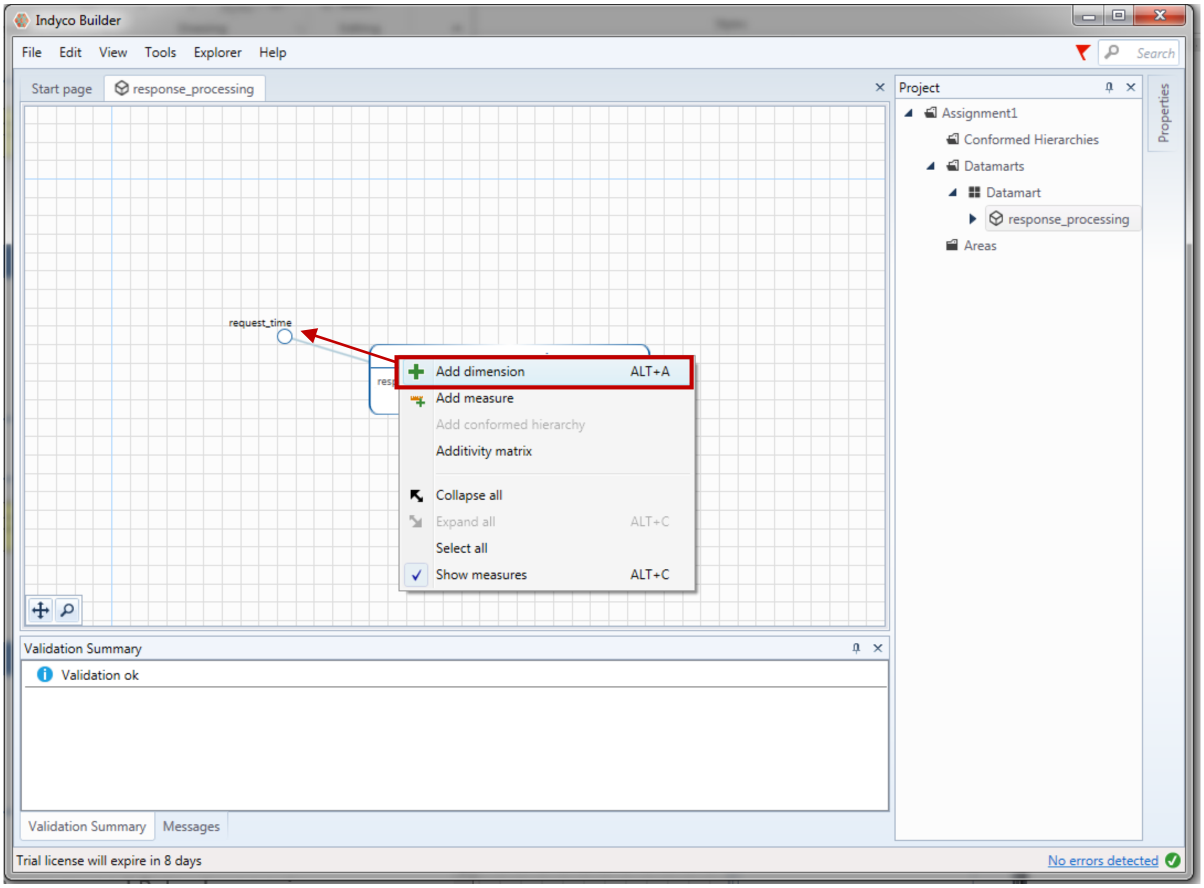
Validation ok

Trial license will expire in 8 days

No errors detected

3. **Adding dimensions.** After adding the focus of the analysis, we need to define the perspectives from which we analyze the added fact (i.e., dimensions).
    a. Following the given requirement, we identify two dimensional concepts from which a user wants to analyze the *processing duration* measure, i.e., *week* and *candidate.*
    b. We can then map them over the given ontology respectively to concepts *year_month_day.year_month_day_weekATRIBUT* and *candidate.candidate_idATRIBUT*.
    c. Additionally, we need to identify in the ontology, how these dimensional concepts are related to the previously found factual concept (i.e., *response_processing*).
    d. If in the ontology the given fact is (by transitivity) related by means of 'to-one' relationship to the given dimensional concept, then we can add the given dimensional concept to the design.
    e. Other concepts identified on the path from the factual to the dimensional concept (i.e., *intermediate concepts*), which do not originate from the given requirement, can potentially play the role of both dimensional and factual concepts. This can therefore generate multiple possible alternative solutions. In addition, each solution must respect the MD integrity constraints, explained in Appendix A.

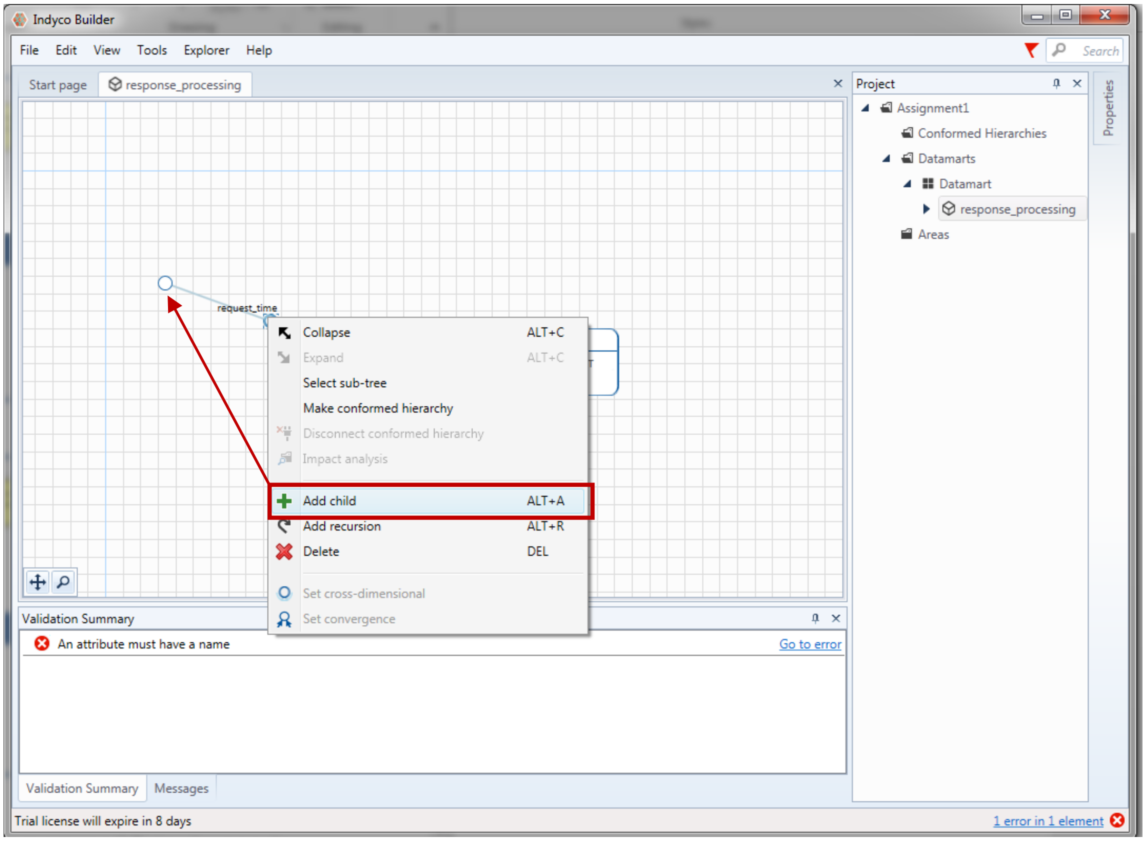
Going from the first solution, where all *intermediate concepts* are interpreted as dimensional concepts, we add them as dimension hierarchies to the previously added fact, as showed in the figures below. Dimension is added by right-click on the fact and choosing *Add dimension*.



Validation ok

Trial license will expire in 8 days

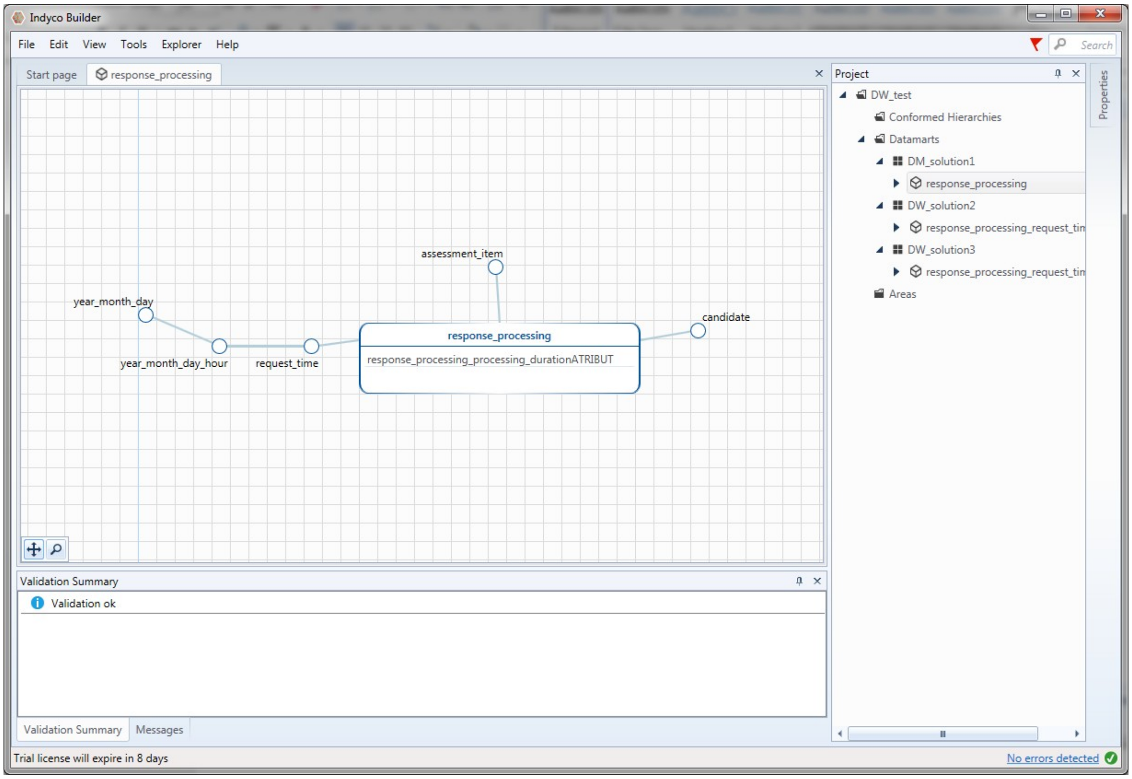
Then each further level of the same dimension hierarchy is added by right-click to the last level and choosing *Add child.*



An attribute must have a name

Trial license will expire in 8 days

Finally, we obtain the first solution of a MD design as shown in the following figure.

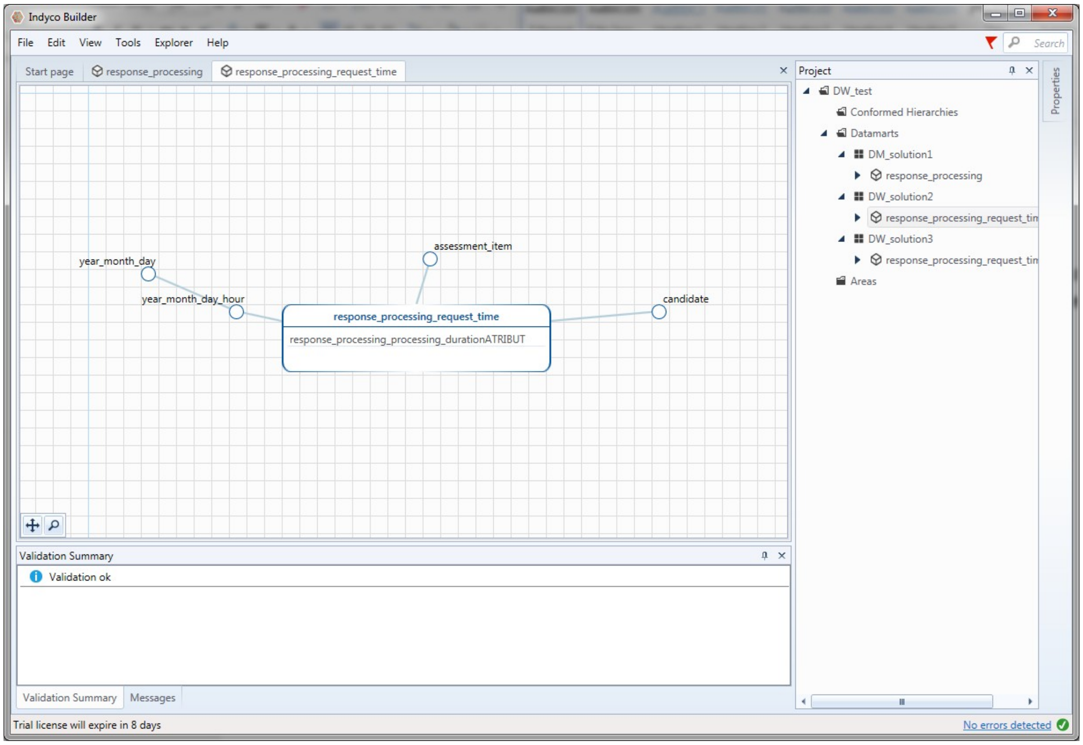


Validation ok

Trial license will expire in 8 days

Figure 2: Solution 1

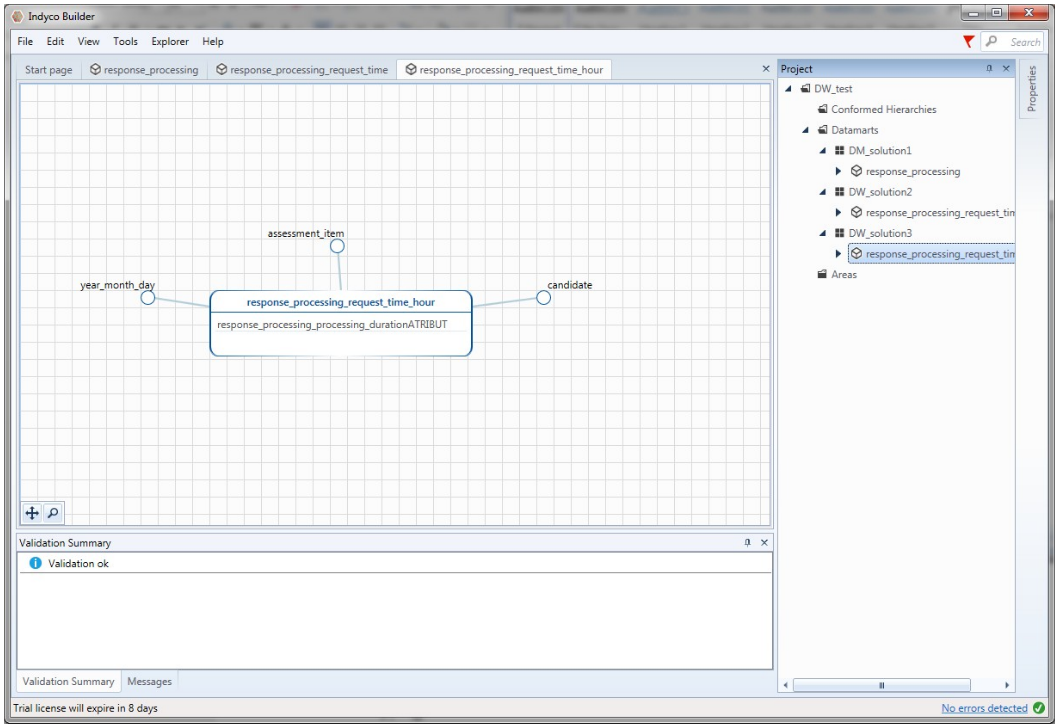
Futhermore, we can obtain two more potential solutions where the *intermediate concepts, i.e., request_time, or request_time* and *year_month_day_hour* play factual roles. This in fact means that the factual concept will be at different levels of granularity (*year_month_day_hour* or *year_month_day*). Notice that the solution where *request_time* plays a dimensional role and *year_month_day_hour* plays a factual role is incorrect as it does not respect the MD integrity constraints (see Appendix A).



Validation ok

Trial license will expire in 8 days

Figure 3: Solution 2



Validation ok

Trial license will expire in 8 days

Figure 4: Solution 3

Appendix A

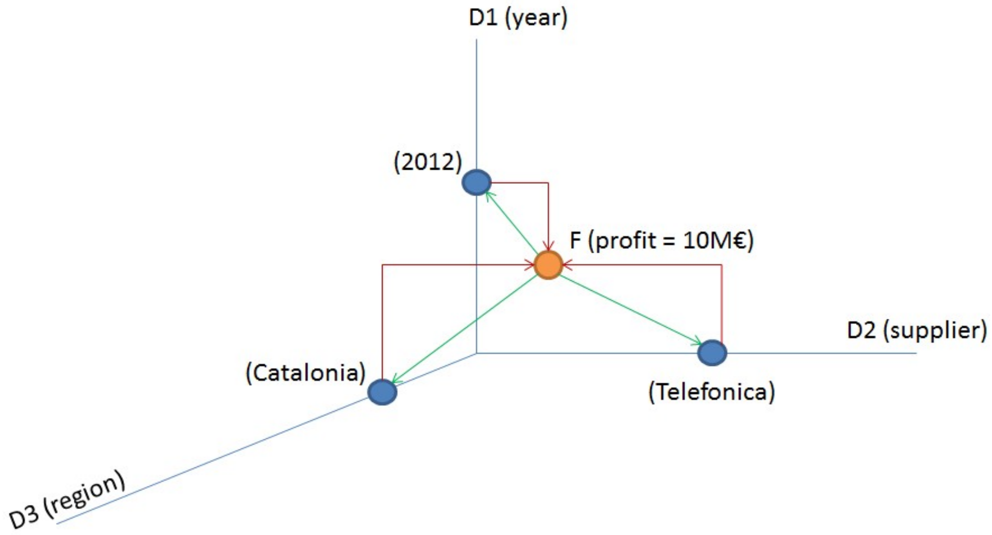
Multidimensional Design - integrity constraints

- MD space: the dimensions over the fact must arrange the correct MD space. On the one side, for each fact instance (e.g., profit = 10M€), there must one and only one dimensional concept in each of the given dimensions (e.g., year = 2012, region = Catalonia, supplier = Telefonica). On the other side, for each set of instances (one from each dimension), there can be one and only one fact instance. See the figure below.
- Summarizability: Data summarization must be correct which is ensured by applying necessary conditions for summarization (see more details in (Mazón, Lechtenbörger, & Trujillo, 2009)):

    1. *Disjointness* - the sets of objects to be aggregated must be disjoint;

    2. *Completeness* - the union of subsets must constitute the entire set; and

    3. *Compatibility* of the dimension - the type of measure being aggregated and the aggregation function.



MD schema solutions

- The produced MD schemas should:
    - **Respect the above MD integrity constraints**, and also
    - Have the **minimal structural complexity** (schema-wise), e.g., number of facts, number of dimensions, attributes, associations, etc.
- To achieve the minimal design we need to look for the parts of the schema that can be:
    - **reused** in answering new requirements
    - correctly **collapsed** with the adjacent parts to build single MD concept

Bibliography

Golfarelli, M., Maio, D., & Rizzi, S. (1998). The Dimensional Fact Model: A Conceptual Model for Data Warehouses. *Int. J. Cooperative Inf. Syst.*, *7*(2-3), 215–247. doi:10.1142/S0218843098000118

Mazón, J.-N., Lechtenbörger, J., & Trujillo, J. (2009). A survey on summarizability issues in multidimensional modeling. *Data Knowl. Eng.*, *68*(12), 1452–1469.