IJISCS (International Journal of Information System and Computer Science)

Full Paper
eISSN 2598-246X
pISSN 2598-0793

APPLYING MULTIDIMENSIONAL CANONICAL PARTITIONING (MCP) APPROACH TO DESIGN URBAN DECISION-SUPPORT SYSTEM

*Corresponding author
apollinaire.batoure@gmail.com

Apollinaire B. Batoure[1], Michel Tchotsoua[2], Kolyang[3]
[1] Department of Computer Engineering, The University of Ngaoundere, Cameroon
[2]Departement of Geography, The University of Ngaoundere, Cameroon
[3]Departement of Computer Science, The University of Maroua, Cameroon

Article history:
Received June 25, 2022
Revised August 26, 2022
Accepted August 29, 2022

Keywords:
Urban transactional system;
Urban decision-support system;
Decision-support system design;
MCP design approach;
Semi-automatic tool Pa-CaM.

Abstract
The design of decision-support systems remains a challenge because no method has yet been approved, although several approaches have been proposed. Multidimensional Canonical Partitioning (MCP) is a supply-driven approach to design decision-support systems, previously proposed. In this approach, decision systems are designed from Entity/Relationship (E/R) schema. The schema provided as input is reduced to a flat relation, if it is not already the case, by using universal relations assumption. The design approach uses two main algorithms. The first is a greedy type heuristic algorithm, and the second a matching algorithm. The approach has six steps. In this paper, we implement algorithms used in the design approach. We then propose a graphical interface as a tool for implementing the approach. The tool, called PaCaM, is intended to be used by decision-support systems designers. We after all apply the design approach on urban transactional systems.

1.0 INTRODUCTION

Computer technologies are nowadays used in all areas of life. The systems implemented through these technologies are consumers and producers of large amounts of data. In addition to their considerable volume, these data are of various sources, types and are produced at an increasingly frequency [1]. All this has limited the possibilities of the previously widely used transactional systems, called On-Line Transactional Processing (OLTP) [2]. This is why On-Line Analytical Processing (OLAP) systems appeared to improve on the weaknesses of transactional systems [3]. While the first one focus on transactional operations, the second focus on operations that can lead to decisions. The two systems cannot dwell in the same environment because they have entirely different objectives and approaches of data management [4]. On the other hand, decision-support systems use transactional systems to implement and even feed themselves. Thus, several research works have focused on the transition from transactional systems and the data they contain, to decision-support systems [5, 6, 7]. Various approaches have been proposed for this purpose. However, it is clear that no single one has yet been approved by everyone (researchers and industrialists) [1, 8]. Also, very few of the proposed approaches go so far, as to provide a tool to accompany implementation.

In previous work [9, 10], the authors proposed a design approach which builds multidimensional data schemas from Entity/Relationship schema. This approach is called Multidimensional Canonical Partitioning (MCP). In this paper, we implement the algorithms used in this design approach. We then propose a graphical interface as a tool for implementing the algorithms. The tool, called PaCaM, is intended to be used by decision-support systems designers. We after all apply the design approach on urban transactional systems. The rest of the paper is structured as follows: Section 2 is dedicated to review on Decision-support systems design; Section 3 recalls the MCP design approach; Section 4 presents urban data and systems we are going to work on; Section 5 focuses on applying MCP on Urban transactional systems, before the conclusion in Section 6.

2.0 REVIEW ON DECISION-SUPPORT SYSTEMS DESIGN

Decision-Support Systems (DSS) design approaches are classified into three main categories, defined and adopted by researchers and industrialists [11]. They are top-down, bottom-up or mixed approaches. The requirement-driven approach is also called user-driven or top-down approach. It defines the conceptual schema of the DSS from user needs. Difficulties with user-driven approaches are instability and constant evolution of user needs. Among top-down approaches, we can mention that one defines by Kimball [12]. This last approach is not formalized but can be seen as a detailed guide for identifying the multidimensional concepts which gives the rise to the DSS schema. In addition to this approach, we have approaches defined by Gam, Mazon, Prat or Tsois [3, 4, 15].

The supply-driven approach is also called data-driven or bottom-up approach. It defines multidimensional schema from operational data sources. Data sources can be found on Entity/Relationship (ER) [16, 17], XML [18] or UML [19] schema. Data-driven approaches directly consider data and operational systems. But if these sources are very numerous, it will require more human, temporal and financial resources for their efficient operating. MCP design approach is of this type [10]. Finally, hybrid, combined or mixed approaches consider user needs and existing data schema to produce multidimensional model [20, 21]. Analysis of sources produced candidate schema and analysis of needs produced ideal schema. Confrontation follows, for production of definitive schema. Among these approaches, we can cite that of Annoni et al. [22] which specifies the requirements in tabular way. From this formalization, an automatic process guides the choice of decision support system architecture. The authors of [22] propose a catalogue of patterns, which capitalizes the development process for the purpose of systematic reuse. We can also mention in this category, approaches proposed by Carneiro, Khouri or Phipps [23, 24]. Works carry out in [9, 10], show that most design approaches have been developed between 1998 and 2010. DSS design are still considered as a research trend. The following works aim at improving the first one or to study other types of sources to consider (i * framework, ontologies, Web...) [23].

Next section will present the MCP design approach.

3.0 MULTIDIMENSIONAL CANONICAL PARTITIONING (MCP) DESIGN APPROACH

In [10], the authors propose a supply-driven approach called Multidimensional Canonical Partitioning (MCP). It takes into account, without distinction, ER data schema. Based on universal relations assumption, the schema is derived into a flatten relation [25]. This step provides a universal relation or a flat schema where all features are grouped into a single entity. This entity is then partitioned vertically, according to characteristics or attributes semantics. To achieve it, we use a greedy type heuristic algorithm. Resulting partitions are candidates for being dimensions in the future DSS data schema. To perform it, we use an algorithm that matches the attributes present in the partitions and those that must actually be in the dimensions. Obtained dimensions are, if necessary, snow flaked (normalized) using the third normal form algorithm (3NF). The central table of the DSS schema, named facts table is generated using guidelines. When all the multidimensional elements are obtained, the data schema is generated using model transformations from QVT (Query View Transformation), in addition to a multidimensional and spatiotemporal design pattern.

The approach is recapped into six steps:

1. Verification of provided ER schema. If it is on universal relation form, we go straight to step 2, otherwise we restructure it according to universal relation assumption;

2. Vertical partitioning, by fragmentation and distribution, of attributes in obtained partitions;
3. Transformation of partitions into dimensions;
4. Normalization of dimensions;
5. Construction of facts table;
6. Generation of multidimensional schema.

Figure 1 presents chart of MCP approach steps. Steps 1 and 4 are principles we used and steps 2, 3, 5 and 6 are principles we proposed in the design approach. R = <C, D> is a relational schema where C is a set of constituents and D, a set of functional dependencies. To coordinate the complete modelling phases (conceptual, logical and physical) of the decision support system, model-driven engineering (MDE) is used. By successive model transformations, one goes from the multidimensional annotation to obtain implementation and ETL (Extract-Transform-Load) codes of the DSS, according to a chosen platform. This part is out of the scope of this paper.

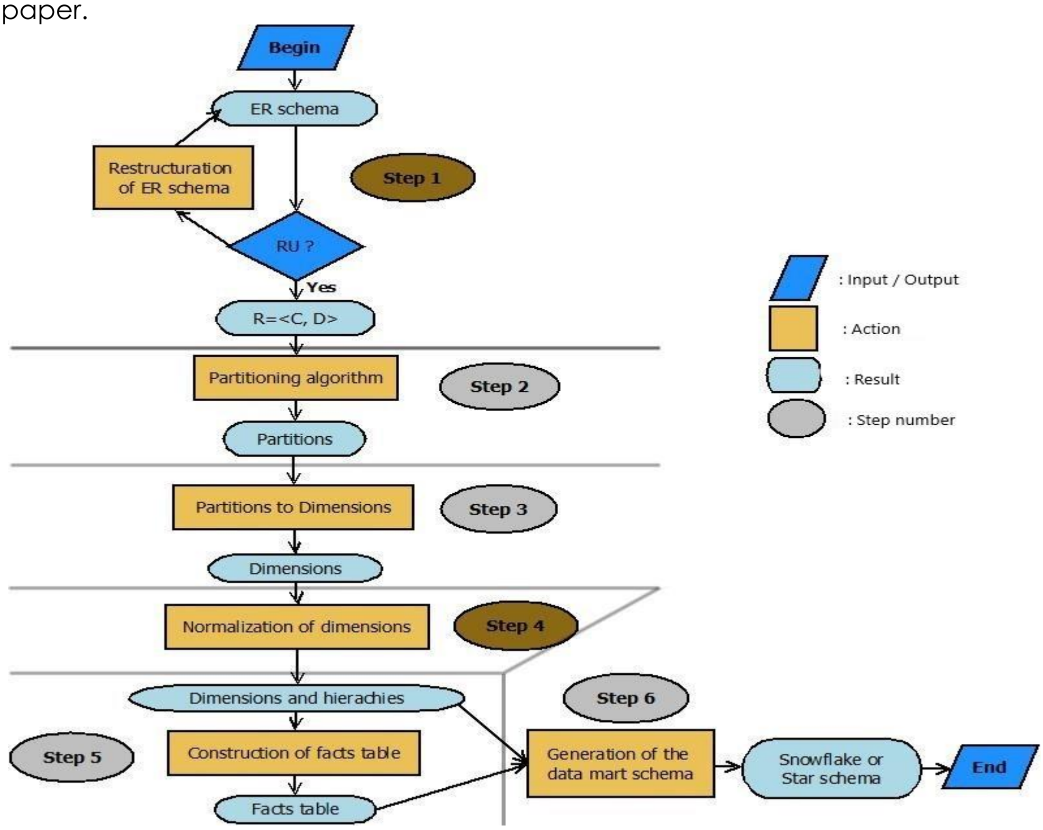
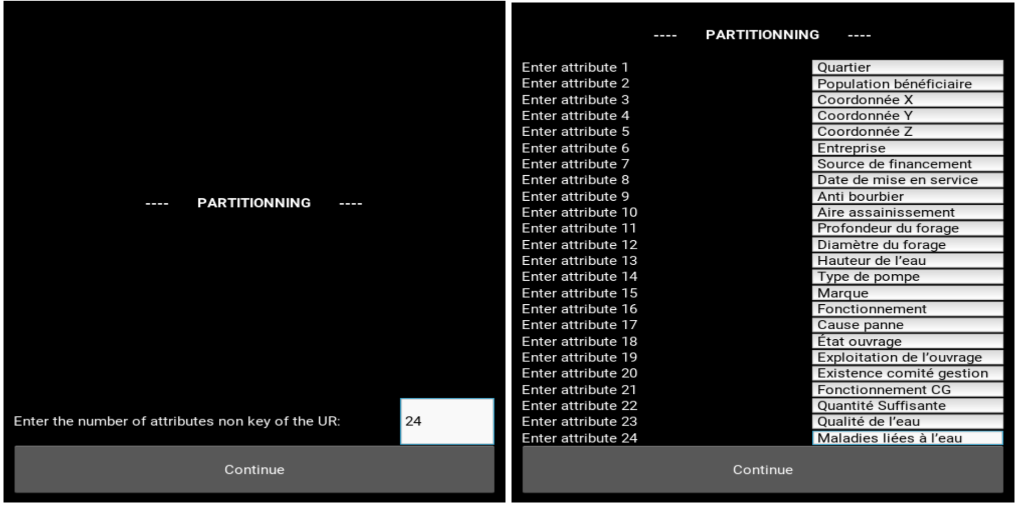


Figure 1. Steps of the design approach

Addition to this implementation, we proposed a graphical interface using the two algorithms. It's a semi-automatic tool that can be used by the designers of the DSS. The interface of the tool is presented in figure 2a and 2b. After entering the number of attributes and the name of each one, interactive part with the designer can start.


---- PARTITIONNING ----

Enter the number of attributes non key of the UR: 24

Continue

---- PARTITIONNING ----

Enter attribute 1	Quartier
Enter attribute 2	Population bénéficiaire
Enter attribute 3	Coordonnée X
Enter attribute 4	Coordonnée Y
Enter attribute 5	Coordonnée Z
Enter attribute 6	Entreprise
Enter attribute 7	Source de financement
Enter attribute 8	Date de mise en service
Enter attribute 9	Anti bourbier
Enter attribute 10	Aire assainissement
Enter attribute 11	Profondeur du forage
Enter attribute 12	Diamètre du forage
Enter attribute 13	Hauteur de l'eau
Enter attribute 14	Type de pompe
Enter attribute 15	Marque
Enter attribute 16	Fonctionnement
Enter attribute 17	Cause panne
Enter attribute 18	État ouvrage
Enter attribute 19	Exploitation de l'ouvrage
Enter attribute 20	Existence comité gestion
Enter attribute 21	Fonctionnement CG
Enter attribute 22	Quantité Suffisante
Enter attribute 23	Qualité de l'eau
Enter attribute 24	Maladies liées à l'eau

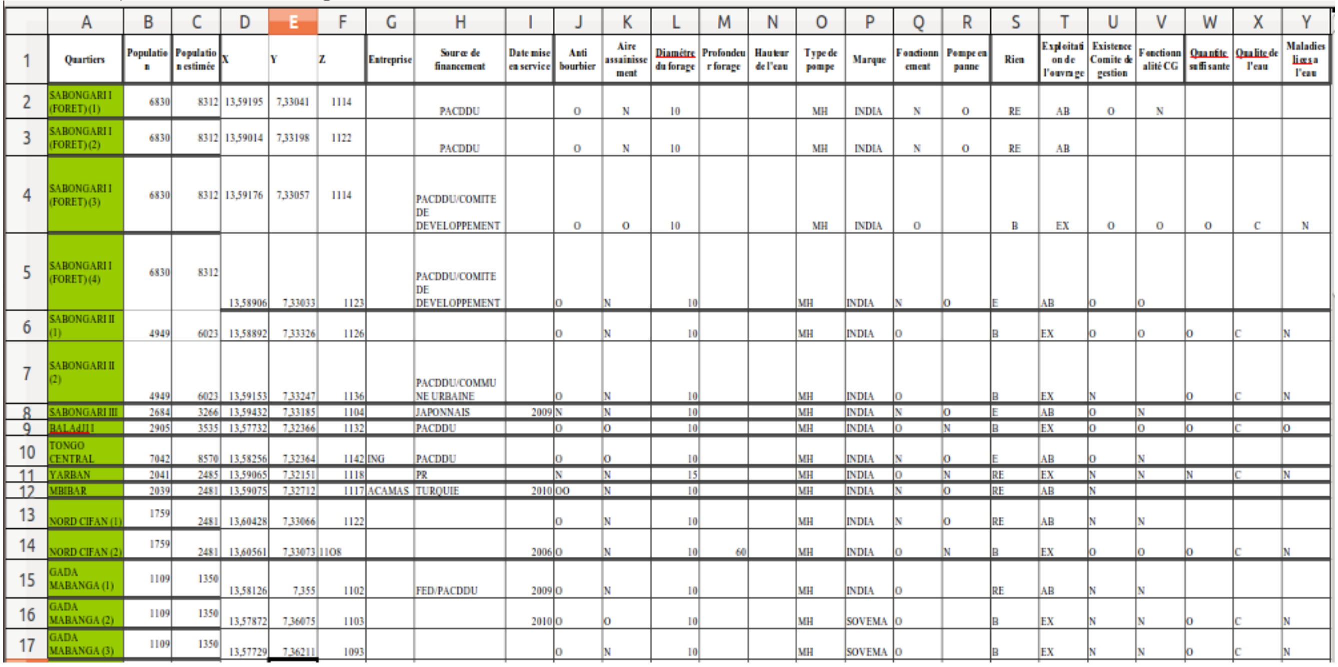
Continue

Figure 2a and 2b. Semi-automatic tool PaCaM

The approach uses two algorithms. They are programmed in the Python language, as well as the PaCaM tool (Figure 2a and 2b). The Python programming language (www.python.org/) was chosen for the implementation of the algorithms. It offers ease and dynamism in the declaration of the data structures we used (set and set of sets). Data manipulation functions are greatly simplified. We also have the facility of assigning mnemonic names to the sets and attributes that we manipulate. All the codes of algorithms, libraries and setup for PaCaM tool are available online through GitHub. After presenting the MCP approach and the PaCaM tool, we will next describe urban transactional systems. It is on those urbans transactional systems that we apply the MCP design approach.

4.0 URBAN TRANSACTIONAL SYSTEMS

The urban transactional systems are the ones we want to transform in order to obtain data schemas for the decision-support systems or Data Warehouse. We focus on some of them. This sample is characteristic of the other. Most Municipality in Cameroon have a Communal Development Plan (CDP). This document contains the projects eligible for funding. It's accompanied by an application called PRO-ADP (PROgiciel d'Aide au Développement Participatif) [26]. From the study made on these urban information systems, we noticed that they are linked to a given urban topic, itself linked to a sectorial. And for this topic or sectorial, all the attributes which characterize it are given in an exhaustive way, all in a single table. All the themes or topics are thus studied and consolidated in a spreadsheet file. Each sheet of the file takes into account a specific topic, without any relationship with other. The topics are, among others, water supply, education, leisure, trade, health or administration. Each sheet has several dozen fields (between 10 and 60) that describe the theme. The example in Figure 3 on the boreholes sheet has 24 fields, in a single table. It is in these spreadsheets that the collected field data are saved. All data include spatial references. The spreadsheet files are then transformed into shape files (.sph) and/or separators files (.csv). The software package of PRO-ADP is implemented using Microsoft ACCESS.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Quartiers	Population	Population estimée	X	Y	Z	Entreprise	Source de financement	Date mise en service	Anti bourbier	Aire assainissement	Diamétre du forage	Profondeur forage	Hauteur de l'eau	Type de pompe	Marque	Fonctionnement	Pompe en panne	Rien	Exploitation de l'ouvrage	Existence Comite de gestion	Fonctionnalité CG	Quantite suffisante	Qualite de l'eau	Maladies liees a l'eau
2	SABONGARI I (FORET)(1)	6830	8312	13,59195	7,33041	1114		PACDDU		O	N	10			MH	INDIA	N	O	RE	AB	O	N			
3	SABONGARI I (FORET)(2)	6830	8312	13,59014	7,33198	1122		PACDDU		O	N	10			MH	INDIA	N	O	RE	AB					
4	SABONGARI I (FORET)(3)	6830	8312	13,59176	7,33057	1114		PACDDU/COMITE DE DEVELOPPEMENT		O	O	10			MH	INDIA	O		B	EX	O	O	O	C	N
5	SABONGARI I (FORET)(4)	6830	8312	13,58906	7,33033	1123		PACDDU/COMITE DE DEVELOPPEMENT		O	N	10			MH	INDIA	N	O	E	AB	O	O			
6	SABONGARI II (1)	4949	6023	13,58892	7,33326	1126				O	N	10			MH	INDIA	O		B	EX	O	O	O	C	N
7	SABONGARI II (2)	4949	6023	13,59153	7,33247	1136		PACDDU/COMMUNE URBAINE		O	N	10			MH	INDIA	O		B	EX	N		O	C	N
8	SABONGARI III	2684	3266	13,59432	7,33185	1104		JAPONNAIS	2009	N	N	10			MH	INDIA	N	O	E	AB	O	N			
9	BALAdJI1	2905	3535	13,57732	7,32366	1132		PACDDU		O	O	10			MH	INDIA	O	N	B	EX	O	O	O	C	O
10	TONGO CENTRAL	7042	8570	13,58256	7,32364	1142	ING	PACDDU		O	O	10			MH	INDIA	N	O	E	AB	O	N			
11	YARBAN	2041	2485	13,59065	7,32151	1118		PR		N	N	15			MH	INDIA	O	N	RE	EX	N	N	N	C	N
12	MBIBAR	2039	2481	13,59075	7,32712	1117	ACAMAS	TURQUIE	2010	OO	N	10			MH	INDIA	N	O	RE	AB	N				
13	NORD CIFAN (1)	1759	2481	13,60428	7,33066	1122				O	N	10			MH	INDIA	N	O	RE	AB	N	N			
14	NORD CIFAN (2)	1759	2481	13,60561	7,33073	11O8			2006	O	N	10	60		MH	INDIA	O	N	B	EX	O	O	O	C	N
15	GADA MABANGA (1)	1109	1350	13,58126	7,355	1102		FED/PACDDU	2009	O	N	10			MH	INDIA	O		RE	AB	N	N			
16	GADA MABANGA (2)	1109	1350	13,57872	7,36075	1103			2010	O	O	10			MH	SOVEMA	O		B	EX	N	N	O	C	N
17	GADA MABANGA (3)	1109	1350	13,57729	7,36211	1093				O	N	10			MH	SOVEMA	O		B	EX	N	N	O	C	N

Figure 3. Data sheet on boreholes

In the framework of a project to create a geo-economic portal for the city of Ngaoundéré, an urban database was developed [27, 28]. The development of the database was carried out in three phases. Using the relational DBMS PostgreSQL, and its spatial extension PostGIS. The geographic database was created and fed with data collected in the field. Then, from a background of satellite images obtained from IKONOS and QUICKBIRD, geo-referenced in Quantum-GIS, the information contained in the database was visualized in the form of layers. To facilitate interaction with the database, a web interface was created using Komposer application. Control of the data entered by users is provided by JavaScript codes. PHP scripts are used to retrieve the data entered from the forms and transfer it to the data-base. The tables in this database have no relationship to each other. They are used for projections in the form of data layers, from coordinates collected in the field, on an image or map background. The tables group together all the information or characteristics on an identified specific theme.

In addition to this works, we can cite that of the Institut Supérieur de Formation et de Recherche Appliquée (ISFRA) of Bamako in Mali, which collected data on the economic activity of Commune V. These data are grouped in a spreadsheet with 25 columns, listing all the information necessary to set up 29,050 activities. Each activity constitutes a record in the file. Based on the Google Earth vector maps, the work of A. Ibrahim in [29], has made it possible to capitalize, in a single tool, all the information on eight (08) Map layers. The layers are integrated into an application called Urban Geographic Information System on Localized Fiscal Resources (SIGU-ReFiLoc).

This presentation, on some urban transactional systems, leads us to identify their specificities. From this study, we note that data collected from the field are integrated into spreadsheets and then projected, from geographic information systems (GIS), onto an image to produce thematic maps. These geographic information systems have summary Data Base Management System (DBMS) that do not, in most cases, implement the relationships between tables [30]. Some urban information systems have an Entity/Association data schema, implemented in a relational DBMS. The DBMS is mainly used for structuring information, facilitating data entry and retrieving information through forms [28]. Thus, even when a DBMS, spatial or not, is used, the entities are not standardized and there are no associations between them. The case of data schemas, where all constituents are grouped into one entity, is of most interest to us. This data schema is a universal relation. It is also referred to as a flat or terraced relationship. Batouré's thesis [31] work provides a table that summarizes the strengths and weaknesses of these urban transactional systems.

We are now going to use the MCP design approach to transform urban transactional systems, into a decisional data schema. In this case, it is the schema of Figure 3, which deals with the boreholes built in the study area.

5.0 USING MULTIDIMENSIONAL CANONICAL PARTITIONING (MCP) ON URBAN TRANSACTIONAL SYSTEMS

To apply MCP design approach, we take the case of the data on the borehole’s topic. The data is collected and entered into a spreadsheet, just like the other themes. It is from this point that the processing and analysis are done. This data source will be used to apply the six steps of the MCP design approach. The schema and data used for applying the design approach are those in the data sheet of figure 3.

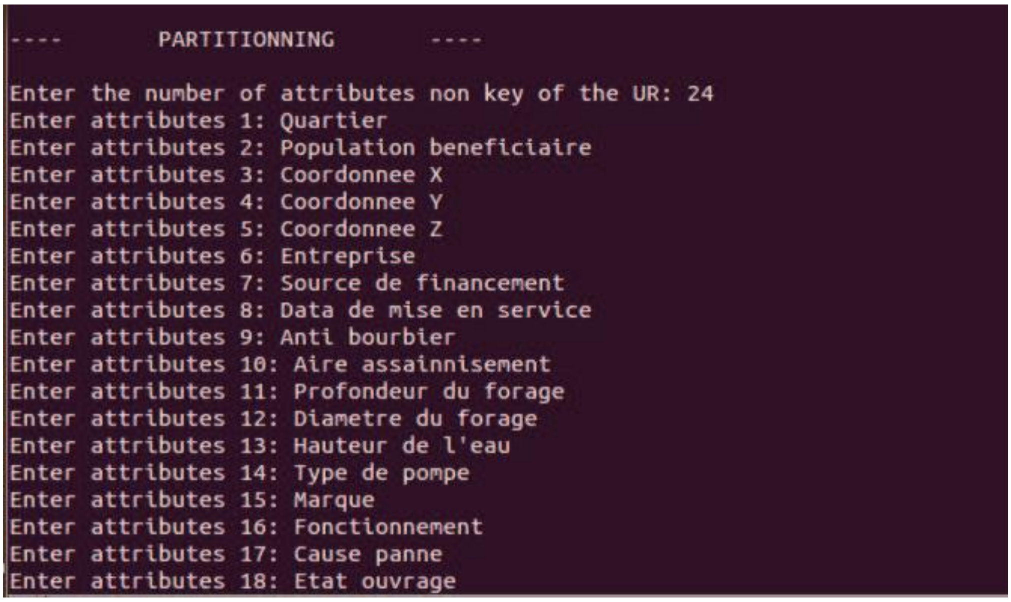
Step 1: Verification of input schema

The input application of the approach is done on any information system with an Entity/Relationship (ER) data schema. The schemas we focus on are generally designed as universal relationships, as in the example in figure 3. This is justified by the description of entities based on field data, without prior sorting. The resulting features or attributes are all collected in a single table. However, if the data schema is not in the form of a universal relation, according to the universal relations assumption [32], it is possible to reduce it to one. For this, projection and join operations are used to obtain a flat relation, taken as a universal relation. Thus, the schema provided as input is in universal relation way. We now move on to the second stage.

Step 2: Partitioning of the universal relation

The table has 24 attributes and is presented as a flat relationship. This data schema is taken as input at this stage. Next, it is a matter of going through the attributes and determining which partition they might describe. The attributes are taken as they are mention in the spreadsheet file. In this case, the set of attributes of the universal relation, taken as input to the partitioning algorithm is:

attributs_RU = {Quartier, Population bénéficiaire, Coordonnée X, Coordonnée Y, Coordonnée Z, Entreprise, Source de financement, Date de mise en service, Anti bourbier, Aire assainissement, Profondeur du forage, Diamètre du forage, Hauteur de l’eau, Type de pompe, Marque, Fonctionnement, Cause panne, État ouvrage, Exploitation de l’ouvrage, Existence comité de gestion, Fonctionnement CG, Quantité Suffisante, Qualité de l’eau, Maladies liées à l’eau}. The algorithm takes as input the total number of attributes and then the exhaustive list of these attributes (Figure 4).

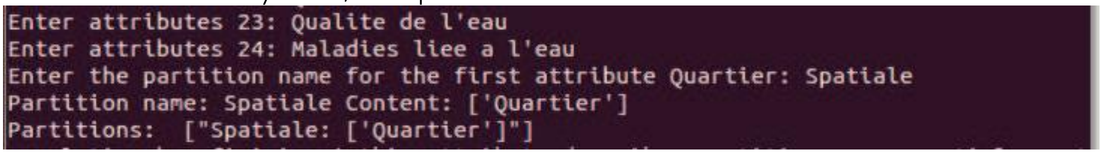


```
----       PARTITIONNING       ----

Enter the number of attributes non key of the UR: 24
Enter attributes 1: Quartier
Enter attributes 2: Population beneficiaire
Enter attributes 3: Coordonnee X
Enter attributes 4: Coordonnee Y
Enter attributes 5: Coordonnee Z
Enter attributes 6: Entreprise
Enter attributes 7: Source de financement
Enter attributes 8: Data de mise en service
Enter attributes 9: Anti bourbier
Enter attributes 10: Aire assainnisement
Enter attributes 11: Profondeur du forage
Enter attributes 12: Diametre du forage
Enter attributes 13: Hauteur de l'eau
Enter attributes 14: Type de pompe
Enter attributes 15: Marque
Enter attributes 16: Fonctionnement
Enter attributes 17: Cause panne
Enter attributes 18: Etat ouvrage
```

Figure 4. Inputs parameters of algorithm

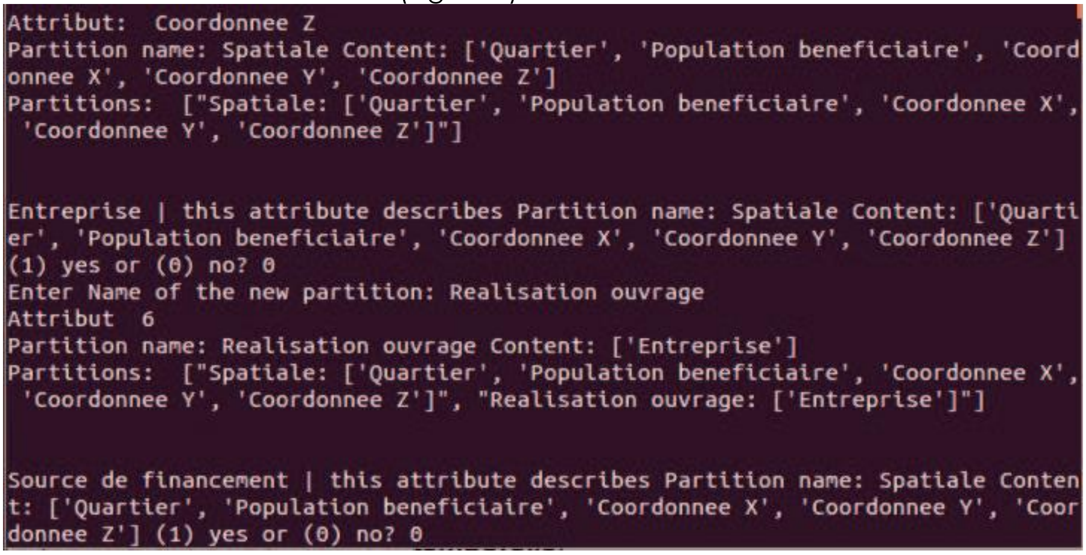
The attributes are browsed one after the other. For the first attribute read, the first partition is created (Figure 5). In this case, the first attribute is “Quartier”. It describes the spatial aspect of the theme under study. Thus, the spatial dimension is created.



```
Enter attributes 23: Qualite de l'eau
Enter attributes 24: Maladies liee a l'eau
Enter the partition name for the first attribute Quartier: Spatiale
Partition name: Spatiale Content: ['Quartier']
Partitions:  ["Spatiale: ['Quartier']"]
```

Figure 5. Creating the first partition

For the following attributes (up to the fifth), they are added in the same partition. The “Entreprise” attribute does not describe the spatial partition. It is therefore necessary to create another partition which describes that attribute and will contain it. The second partition is thus created, namely “Realisation ouvrage”. In the same way, the “Source de financement” attribute is added to this dimension (Figure 6).



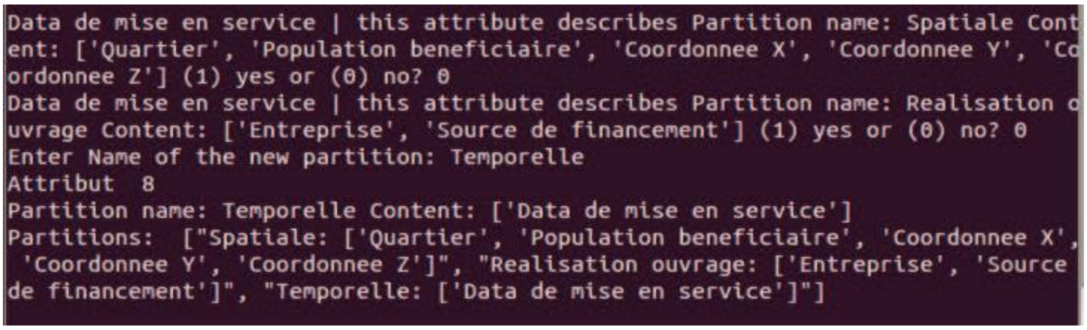
```
Attribut:  Coordonnee Z
Partition name: Spatiale Content: ['Quartier', 'Population beneficiaire', 'Coord
onnee X', 'Coordonnee Y', 'Coordonnee Z']
Partitions:  ["Spatiale: ['Quartier', 'Population beneficiaire', 'Coordonnee X',
 'Coordonnee Y', 'Coordonnee Z']"]


Entreprise | this attribute describes Partition name: Spatiale Content: ['Quarti
er', 'Population beneficiaire', 'Coordonnee X', 'Coordonnee Y', 'Coordonnee Z']
(1) yes or (0) no? 0
Enter Name of the new partition: Realisation ouvrage
Attribut  6
Partition name: Realisation ouvrage Content: ['Entreprise']
Partitions:  ["Spatiale: ['Quartier', 'Population beneficiaire', 'Coordonnee X',
 'Coordonnee Y', 'Coordonnee Z']", "Realisation ouvrage: ['Entreprise']"]


Source de financement | this attribute describes Partition name: Spatiale Conten
t: ['Quartier', 'Population beneficiaire', 'Coordonnee X', 'Coordonnee Y', 'Coor
donnee Z'] (1) yes or (0) no? 0
```

Figure 6. Creating a new partition

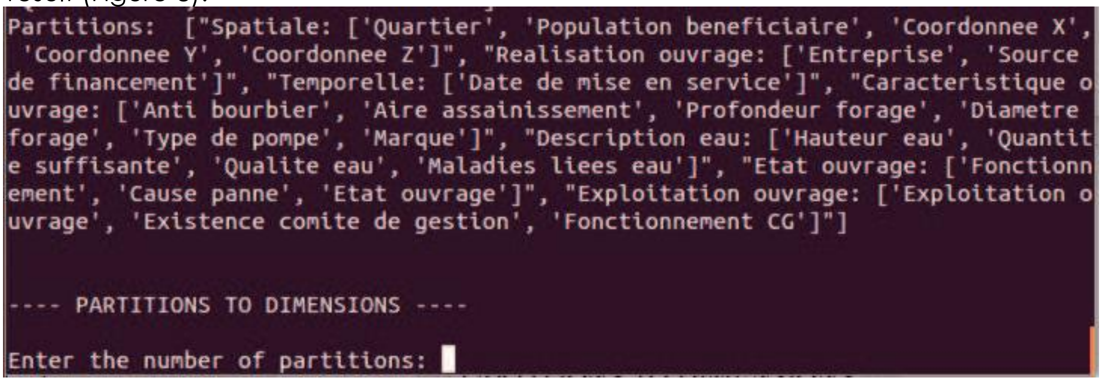
The “Date de mise en service” attribute leads us to create a new partition, the temporal one (Figure 7).



```
Data de mise en service | this attribute describes Partition name: Spatiale Cont
ent: ['Quartier', 'Population beneficiaire', 'Coordonnee X', 'Coordonnee Y', 'Co
ordonnee Z'] (1) yes or (0) no? 0
Data de mise en service | this attribute describes Partition name: Realisation o
uvrage Content: ['Entreprise', 'Source de financement'] (1) yes or (0) no? 0
Enter Name of the new partition: Temporelle
Attribut  8
Partition name: Temporelle Content: ['Data de mise en service']
Partitions:  ["Spatiale: ['Quartier', 'Population beneficiaire', 'Coordonnee X',
 'Coordonnee Y', 'Coordonnee Z']", "Realisation ouvrage: ['Entreprise', 'Source
de financement']", "Temporelle: ['Data de mise en service']"]
```

Figure 7. Temporal partition

The algorithm continues to run for each attribute read. And, whenever the read attribute de-scribes an existing partition, this attribute is added to the partition. Otherwise, a new partition is created, which will include it. We have the set of partitions and their attributes as the final result (Figure 8).



```
Partitions:  ["Spatiale: ['Quartier', 'Population beneficiaire', 'Coordonnee X',
 'Coordonnee Y', 'Coordonnee Z']", "Realisation ouvrage: ['Entreprise', 'Source
de financement']", "Temporelle: ['Date de mise en service']", "Caracteristique o
uvrage: ['Anti bourbier', 'Aire assainissement', 'Profondeur forage', 'Diametre
forage', 'Type de pompe', 'Marque']", "Description eau: ['Hauteur eau', 'Quantit
e suffisante', 'Qualite eau', 'Maladies liees eau']", "Etat ouvrage: ['Fonctionn
ement', 'Cause panne', 'Etat ouvrage']", "Exploitation ouvrage: ['Exploitation o
uvrage', 'Existence comite de gestion', 'Fonctionnement CG']"]


---- PARTITIONS TO DIMENSIONS ----

Enter the number of partitions: 
```

Figure 8. Resulting set of partitions

At the end of the step, we have seven partitions in which all the attributes are distributed. An attribute can only be found in one partition at a time. If there is any ambiguity about an attribute, the designers of the original database should be consulted, and if necessary, the end-users or decision-makers. They are in a position to explain the actual semantic of each attribute.

These partitions and the attributes they contain are input for the third step. The spatial and temporal dimensions are pre-designed according to the archetype of the design pattern profile [33]. In step 3, we will focus on the remaining five dimensions.

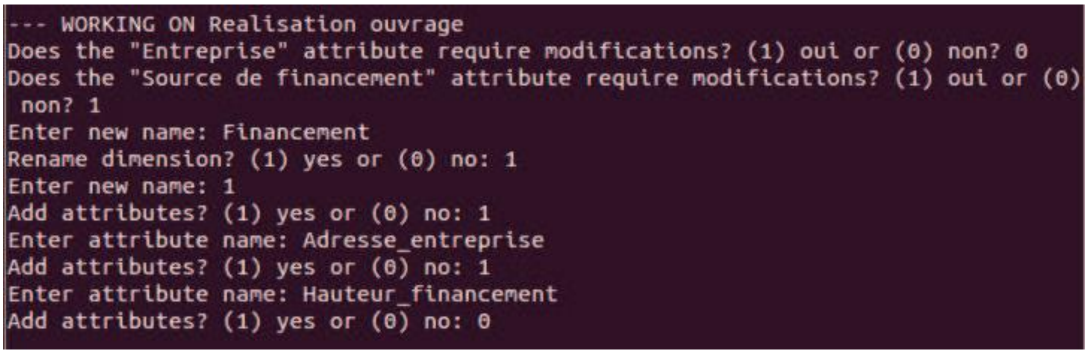
Step 3: Processing partitions into dimensions

In this step, the previously obtained partitions are transformed into dimensions for our future multidimensional data schema. This means that each attribute in the partition, is kept as its, modified nor abandoned. For this purpose, we use Table 1 structure.

Table 1. Processing of Partitions into Dimensions

		Partition	
	Attribute 1	*Attribute 2*	*Attribute 3*
Dimension	**OK**	**MOD**	**ABD**

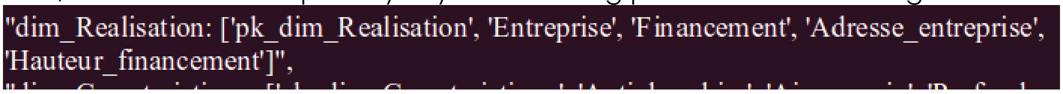
OK: The attribute is taken as it is; **MOD**: The attribute undergoes one or more modifications; **ABD**: The attribute is not taken into consideration in the dimension. It is therefore a question of producing such a table for each partition we have and their attributes. Afterwards, we can rename or not the obtained dimension; add possible attributes, according to the real needs; necessarily add a primary key to it and include the obtained dimension in the set of dimensions. With the previously obtained set of partitions as an input parameter, the second algorithm will be executed. Execution on the partition "Realisation ouvrage", for example, leads us to results shown in Figure 9.



```
--- WORKING ON Realisation ouvrage
Does the "Entreprise" attribute require modifications? (1) oui or (0) non? 0
Does the "Source de financement" attribute require modifications? (1) oui or (0)
 non? 1
Enter new name: Financement
Rename dimension? (1) yes or (0) no: 1
Enter new name: 1
Add attributes? (1) yes or (0) no: 1
Enter attribute name: Adresse_entreprise
Add attributes? (1) yes or (0) no: 1
Enter attribute name: Hauteur_financement
Add attributes? (1) yes or (0) no: 0
```

Figure 9. Processing partition "Realisation ouvrage" to dimension

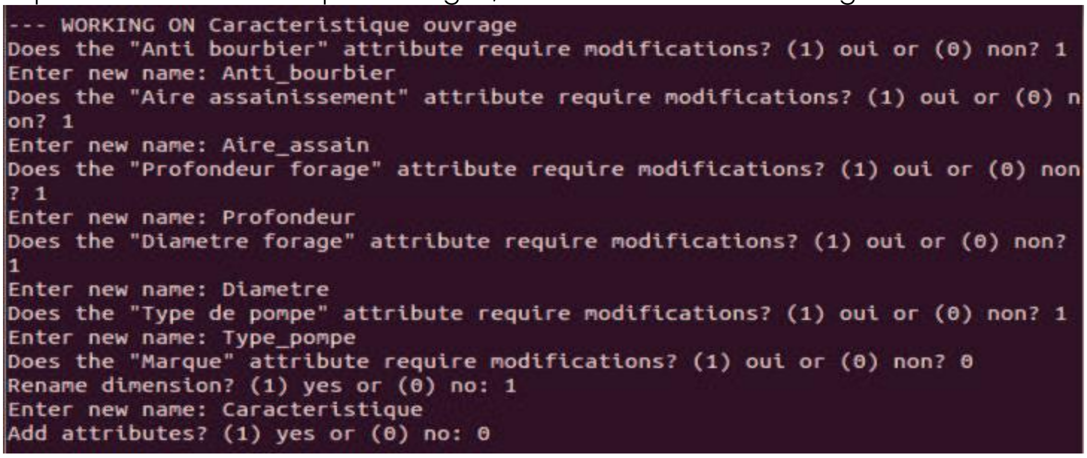
The dimension is renamed, as are some of its attributes. Two attributes are added to the dimension, in addition to the primary key. The resulting partition is shown in Figure 10.



```
"dim_Realisation: ['pk_dim_Realisation', 'Entreprise', 'Financement', 'Adresse_entreprise',
'Hauteur_financement']",
```

Figure 10. Dimension "Realisation"

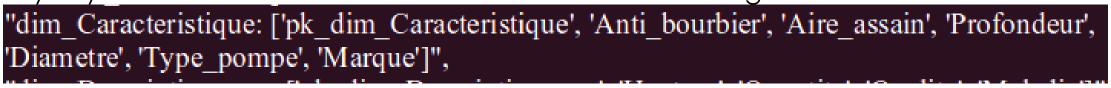
For the partition "Caractéristique ouvrage", execution is as shown in Figure 11.



```
--- WORKING ON Caracteristique ouvrage
Does the "Anti bourbier" attribute require modifications? (1) oui or (0) non? 1
Enter new name: Anti_bourbier
Does the "Aire assainissement" attribute require modifications? (1) oui or (0) n
on? 1
Enter new name: Aire_assain
Does the "Profondeur forage" attribute require modifications? (1) oui or (0) non
? 1
Enter new name: Profondeur
Does the "Diametre forage" attribute require modifications? (1) oui or (0) non?
1
Enter new name: Diametre
Does the "Type de pompe" attribute require modifications? (1) oui or (0) non? 1
Enter new name: Type_pompe
Does the "Marque" attribute require modifications? (1) oui or (0) non? 0
Rename dimension? (1) yes or (0) no: 1
Enter new name: Caracteristique
Add attributes? (1) yes or (0) no: 0
```

Figure 11. Processing the partition "Caracteristique ouvrage"

The dimension and attributes are renamed. No other attribute is added, except for the primary key. The final result for this transformation is shown in Figure 12.



```
"dim_Caracteristique: ['pk_dim_Caracteristique', 'Anti_bourbier', 'Aire_assain', 'Profondeur',
'Diametre', 'Type_pompe', 'Marque']",
```

Figure 12. Dimension "Caracteristique"

The transformation into dimensions of the other partitions follows the same process. At the end of this step, we have all the dimensions. They constitute the analysis axes of the decision-making system. For the dimensions that require it, they can be normalized in order to obtain a better description of the data.

Step 4: Normalization of dimensions

Snowflaking, or normalization of dimensions, is used to give more precisions to analysis axes, also called dimensions. The more precisions they are, the finer and more detailed are the analysis. The operation is based on third normal form (3NF). The precondition for universal relations assumption is: "two attributes with the same name represent the same concept" or "two attributes representing distinct concepts have different names". This assumption makes it possible to uniquely name an attribute, without any ambiguity. This leads us to obtain the first normal form (1NF) in the dimensions [40]. Moreover, as the primary keys of the tables are not composite, we take for granted the second normal form (2NF).

For the dimension "Caracteristique" for example, the attributes group {anti_bourbier, aire_assain, profondeur, diametre} depend transitively, by the primary key {cle_caracteristique}, to the attributes group {type_pompe, marque}. The breakdown allows us to give more details to the type of attribute "Pompe" used. By adding foreign key, we have the two schemas below: "Caracteristique" {cle_caracteristique, anti_bourbier, aire_assain, profondeur, diametre, cle_pompe} and "Pompe" (cle_pompe, type_pompe, marque, cout}. Likewise, for the dimension "Description_eau", it can be breakdown to obtain the two schemas bellow, where we give more details on illness related to water "Description_eau" {cle_description, hauteur, quantite, qualite, cle_maladie} and “Maladie” {cle_maladie, type, agent_pathogene, frequence}. Other dimensions, if necessary, can also been breakdown.

Step 5: Construction of the facts table

The construction of the facts table is carried out by guidelines. The primary keys of the dimensions directly related to it are added to that table. All these keys together form the primary key of the facts table. Subsequently, we can add all the desired measures. They can be, according to the meta-model of the spatiotemporal profile, thematic, spatial, temporal or spatiotemporal [33]. They are defined by the needs or requirements of users and decision makers.

Application of the guidelines to this example gives us the following scheme:

“Faits_forage » (cle_description, cle_caracteristique, cle_exploitation, cle_etat,
cle_realisation, cle_spatiale, cle_temporelle, forage/unite_spatiale, forage/unite_temporelle,
forage/unite_spatiale/unite_temporelle, forage/etat/unite_spatielle,
forage/realisation/unite_temporelle);

As well as we have the seven dimensions around the facts table, it’s possible to have 27 (128) views constituting measures or indicators. The number becomes even larger if we have to take into consideration the granularities of the dimensions. At this level of the design approach application, we have all the necessary elements to generate the multidimensional data schema.

Step 6: Generating the multidimensional data schema

From the multidimensional elements (dimensions, granularities, facts table, measures...) obtained in the previous steps, the data schema can be produced. For this purpose, both the profile of the multidimensional and spatiotemporal design pattern, in addition to the archetype derived from it, are used []. This archetype is the generic prototype on which the data schema to be produced is modelled on. Model transformations are also used to move from the multidimensional annotation model (**MetaModelAM**) to the data mart model (**MetaModelMD**), as defined in the Computational level (CIM) actions of Model-Driven Engineering (MDE) [34]. These transformations allow us to specified which multidimensional element is related to which one, and with which role to play in the overall scheme of the decision-support system. The resulting global schema (Figure 13) is snowflake-like, as some dimensions are derived in granularities.

We thus started from the universal data schema (Figure 3) to end to the multidimensional data schema (Figure 13) based on the Multidimensional Canonical Partitioning (MCP) design approach. Now that the data schema has been determined, next steps are to implementing and operating the system.

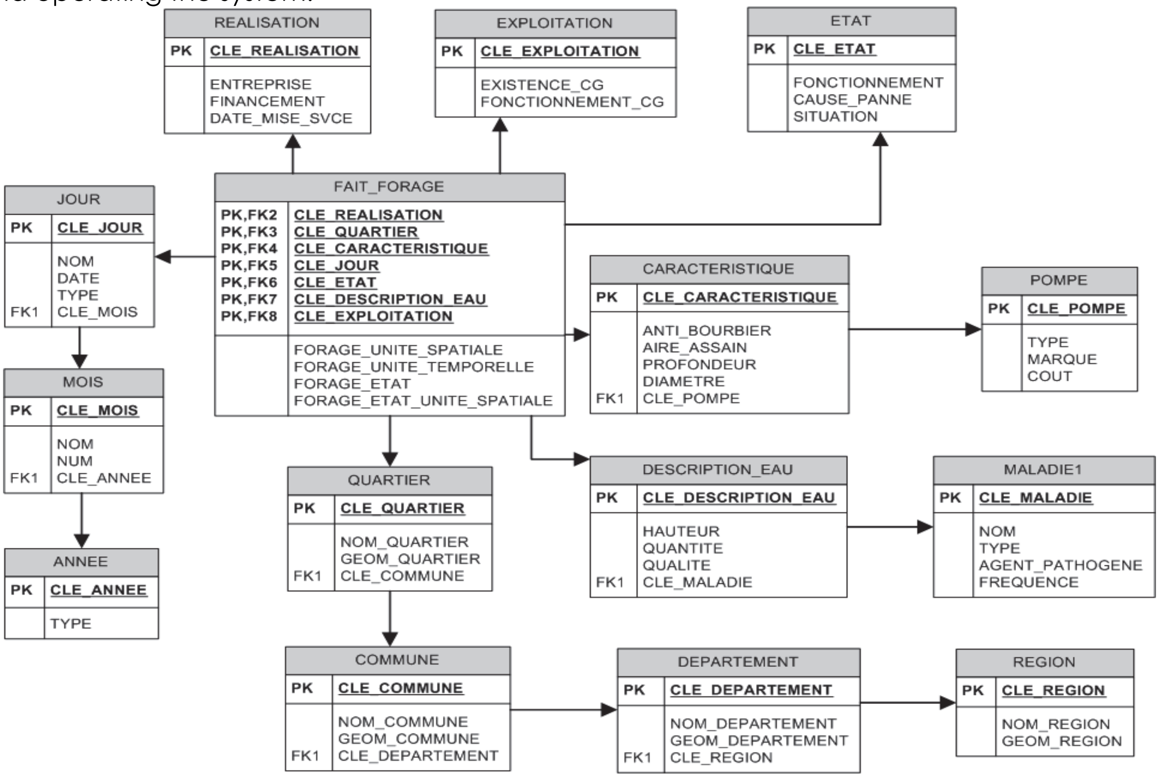


Figure 13. Multidimensional data schema

6.0 CONCLUSION

In this work, we have applied the Multidimensional Canonical Partitioning (MCP) design approach to urban transactional systems, in order to transform them into a decision-support system (DSS). This transformation makes it possible to implement a real (DSS) and to benefit from the advantages of big data technology. Now that the design has been carried out on this example, of the borehole's topic, it is therefore a matter of doing the job for all the other topics (more than 60 topics) on one hand; and on the other hand, moving forward to the implementation stages of the whole data schema and then running of the (DSS) obtained. The implementation of the (DSS) will be done using model-driven engineering (MDE).

REFERENCES

[1] Golfarelli, M., Rizzi, S. 2018. From star schemas to big data: 20+ years of data warehouse research. In: A comprehensive guide through the Italian database research over the last 25 years. Springer International Publishing, pp 93–107, 2018.

[2] Claybrook, B. 2014. OLTP (online transaction processing). Wiley Publishing.

[3] Valencia-García, R., Paredes-Valverde, M.A., Salas-Zárate, M.P., Alor-Hernández, G. 2018. Exploring Intelligent Decision Support Systems. Current State and New Trends, Springer.

[4] Prakash N., Prakash, D. 2018. Data Warehouse Requirements Engineering. A Decision Based Approach, Springer Nature Singapore Pte Ltd.

[5] Cravero A. and Sepulveda, S. 2016. Methodologies, techniques and tools for OLAP design: A Systematic Mapping Study. IEEE Latin America Transactions, 14(2) :943–950.

[6] Vaisman, A. and Zimányi, E. 2014. A Method for Data Warehouse Design. In Data Ware-house Systems, chapter 1. Springer-Verlag Berlin Heidelberg, Berlin.

[7] Chandra, P., Gupta, M. K. 2017. Comprehensive survey on data warehousing research. Int. j. inf. technol., Springer, 8p.

[8] Batoure, A.B., Kolyang. 2018. Designing Decision Support Systems Approaches using Entity/Relationship Data Schema: A Survey, International Journal of Scientific & Engineering Research Volume 9, Issue 9, 308-314.

[9] Arfaoui, N. and Akaichi, J. 2012. Data Warehouse: Conceptual and Logical Schema - Survey. International Journal of Enterprise Computing and Business Systems (IJECBS), 2(1) :31.

[10] Batoure, A.B., Kolyang and Tchotsoua, M. 2017. Using Multidimensional Canonical Partitioning (MCP) as a Supply-Driven Approach for Data Warehouses Design. International Journal of Computer (IJC), 25(1):52–62.

[11] Cuzzocrea, A., Bellatreche, L. and Song, Il Y. 2013. Data Warehousing and OLAP over Big Data: Current challenges and future research directions. In DOLAP'13, pages 67–70, San Francisco, CA, USA.

[12] Kimball, R. and Ross, M. 2011. The data warehouse toolkit: the complete guide to dimensional modelling. Wiley, 2nd edition.

[13] Piattini, M., Mazón, J.-N., Trujillo, J., Serrano, M. 2005. Designing data warehouses: from business requirement analysis to multidimensional modeling. Rebnita, 5 :44 – 53.

[14] Gam, I. and Salinesi, C. 2006. Analyse des Exigences pour la Conception d'Entrepôts de Données. In INFORSID, pages 1023–1038.

[15] Tsois, A., Karayannidis, N. and Sellis, T. K. 2001. MAC: Conceptual data modelling for OLAP. In Proceedings of the 3rd Intl. Workshop on Design and Management, volume 39.

[16] Saxena, G. 2014. Data Warehouse Designing: Dimensional Modelling and E-R Modelling. International Journal of Engineering Inventions, 3(9) :28–34.

[17] Chen, Y.-T. and Hsu, P.-Y. 2005. An Efficient and Grain Preservation Mapping Algorithm: From ER Diagram to Multidimensional Model. Issads 2005, Lncs 3563, pages 331–346.

[18] Chandwani, G. 2014. Designing of Star Schemas from XML Document. International Journal of Multidisciplinary and Current Research, 2 :10.

[19] Pahwa, P., Taneja, S. and Jain, S. 2011. Design of a Multidimensional Model Using Object Oriented Features in UML. International Research Journal, 1(1) :14.

[20] Mankad, D., Dholakia, P. 2013. The Study on Data Warehouse Design and Usage, International Journal of Scientific and Research Publications, Volume 3, Is-sue 3.

[21] Vaisman, A. and Zimányi, E. 2014. A Method for Data Warehouse Design. In Data Warehouse Systems, chapter 1. Springer-Verlag Berlin Heidelberg, Berlin.

[22] Annoni, E., Ravat, F., Teste, O. and Zurfluh, G. 2006. Automating the choice of decision support system architecture. Lecture Notes in Computer Science, 4080 :244–253.

[23] Khouri, S., Bellatreche, L., Jean, S. and Ait-Ameur, Y. 2014. Requirements driven data warehouse design: We can go further. In Lecture Notes in Computer Science, volume 8803, pages 588–603.

[24] Phipps, C. and Davis, K. C. 2002. Automating data warehouse conceptual schema design and evaluation. Dmdw, pages 1–10.

[25] Leymann, F. 1989. "A Survey of the Universal Relation Model." Data & Knowledge Engineering 4, pp. 305-320.

[26] ACAGER and PNDP. 2013. Plan de développement communal de Ngaoundéré 2ème. Technical report, ACAGER, PNDP.

[27] Batouré, A. B. 2011. Informatique appliquée aux SIG en ligne : Création de la base de données urbaines de Ngaoundéré. Éditions Universitaires Européennes (EUE).

[28] Batouré, A. B. and Tchotsoua, M. 2017. Usage des logiciels libres pour la gestion des données à référence spatiale. Afrique Science, 13(3) :377 – 389.

[29] Ibrahim, A. 2015. Géomatique et collecte des recettes fiscales dans la Commune V du District de Bamako. Master Professionnel, Université de Ngaoundéré.

[30] Messina, C. N. 2011. Diagnostique du SIG de la Communauté Urbaine de Yaoundé (CUY) et proposition des grandes orientations en vue de l'élaboration d'un schéma directeur (CUY) 2012-2017. Master Professionnel, Université de Douala.

[31] BATOURE, A.B. 2019. Modelling a decision-making system from transactional systems of the universal relationship type. PhD thesis in Computer Engineering. The University of Ngaoundéré.

[32] Harihodin, S. 1992. "A Universal Relation Approach for Natural Query In Logic Database System," Jurnal Teknologi, vol. 20, pp. 46-64.

[33] Batoure, A.B. and Kolyang. 2020. Multidimensional and SpatioTemporal Design Pattern for Decision-Support Systems Schema Generation, International Journal of Computer Applications Technology and Research Volume 9–Issue 06, 211-216.

[34] Batoure, A.B. and Kolyang. 2020. Using Model-Driven Engineering for Decision Support Systems Modelling, Implementation and Powering, International Journal of Data Engineering (IJDE), Volume (8): Issue (1).