International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 2 Issue 5 ǁ May. 2013 ǁ PP.74-80

Personalization and Recommendation of Queries in Multidimensional Data Base

Nesrine Chaibi[1], Mohamed Salah Gouider[2]

[1] *University of Gabès. Higher Institute of Management of Gabès - Tunisia*
[2] *University of Tunis. Higher Institute of Management of Tunis -Tunisia*

ABSTRACT: *The Multidimensional Data Base provides the means for analyzing large volumes of data. However, the result of current query can be imperfect and obsolete and disturb users to take decision. Therefore, we propose to customize the results returned to the user by integrating current constraints (absolute and contextual preferences) in the process of data extraction and recommend queries from those recorded in the log file to improve their navigation. Moreover, we propose to recommend queries that seem relevant to that user thanks to queries stored in the log file. To evaluate the performance of two algorithms proposed, we implemented a prototype java above the MySQL DB. Also, we have created a database for the library of our university and we used the open source Mondrian server by taking advantage of the multidimensional database analysis of sales 'FoodMart' which is installed with OLAP Services. Moreover, we have carried out experiments to demonstrate the effectiveness and achievability of the proposed algorithms to help user to refine the size of the data returned by his query and to predict its future queries.*

Keywords - *personalization, recommendation, contextual preference, absolute preference, queries of log file.*

I. INTRODUCTION

Although Data Warehouses have emerged to implement an information system dedicated to strategic decisions and to define queries on large data volumes [12], adjusting query results in the specific needs to each decision maker and the restriction of results to the most relevant data remains a major problem [11].
Also, the decision maker is obliged to express several queries to obtain a result as close as possible to their needs. Thus, the search for information in a MDB takes place in several steps and often requires a significant effort [13]. However, the decision maker, who is generally a non computer, is not often able to perfectly master the MDB scheme that incorporates several complex multidimensional structures [11]. Therefore, several approaches of personalization and recommendation queries in data warehouse appeared to meet the expectations of the typical decision makers and facilitate their analysis.

The paper is organized as follows: Section 2 briefly reviews the related works. Section 3 introduces the basic definitions for OLAP data modeling and the description of user model. In Section 4, we present and describe the proposed algorithms for personalization and recommendation in MDBs. Finally, Section 5 presents our experimental results. We conclude and present future works in Section 6.

II. RELATED WORKS

Several approaches of personalization and recommendation have emerged to help decision makers to explore the large volume of data in MDBs. A benchmarking study of OLAP personalization methods is proposed in [1]. Generally personalization approaches can be classified in two classes: one is based on the content [7, 14, 15] and the other is a hybrid [3, 4].

Personalization based on the content takes into accounts the needs and preferences of users. Preferences and knowledge are represented in the form of orders (qualitative representation of preferences), in the form of rules such as "if-then" [7] or ECA (Event, Condition, Action) rules to generate multidimensional tables containing only the data identified as relevant based on the weight set by the user [15]. Furthermore, in [14], the authors calculated the degree of interest (scores) of preferences in relation to CAC (Current Analysis Context) to integrate the K best preferences in the initial query. In this case, it appears that the integration score can enhance or devalue a preference, as a contextual preference may cover more the CAC than absolute preference. This does not consider the absolute preferences with the highest priority.

Hybrid personalization combines the personalization of content and the viewing query results within the constraints of display to better match the schema of the data warehouse needs of the user. Thus, the techniques of information filtering based on user profile are used to customize their queries. Then the constraints explored visualization to present the best result [3, 4].

The second technique exploited in the field of personalization in the Data Warehouse is the recommendation. This approach can be classified into two categories: (i) methods based on the content [5, 14]

and (ii) those based on collaborative filtering [6, 2, 8, 9, 10]. The principle of the first method is to recommend to the current user queries similar to those which interested him in the past. In [5], Bentayeb et al have defined an operator called RoK OLAP which is based on the combination of traditional operator and Roll-Up Method for Clustering or recommend OLAP queries based on the operation of the user's profile. Recommendation based on collaborative filtering is to recommend to the current user a set of queries based on the browsing history of this or other users using the Levenshtein distance (between sessions) and the Hausdorff distance (between queries) [8, 9, 13, 14, 15]. In addition, Colas et al proposed to organize the query log of an OLAP server in the form of a website [6] to give account of what was done in previous analysis sessions. But if the log is large, browsing this website can be tedious. For that, they proposed another method to summarize and examine logs OLAP queries [2]. Also to recommend queries, Giacometti et al [8, 9, 10] and Nègre [15] compute a similarity between the sequence of current user queries and the sequence of previous queries from other users that are saved by the server. In [8], they grouped queries to detect whether the current session is a prefix of an existing session by identifying the position (p) where there is agreement and recommended queries located after this position. In [15], Nègre used the Hausdorff distance for grouping queries into classes where each one is a set of queries close together. Then she sought the positions of p where the current generalized session is a subsequence of each generalized previous sessions. From these classes, she generated all candidate queries that are ordered according to their adequacy with the current user's profile.

III. BASIC DEFINITIONS

OLAP data is usually stored in fact tables which are composed of attributes of measures and attributes of dimensions that are connected to this fact.

A multidimensional database is defined by MDB = (F^{MB}, D^{MB}) où F^{MB} = { $F_1, \dots, F_n$} is a set of facts and D^{MB} = {$D_1, \dots, D_n$} is a set of dimensions.

- For $\forall i \in [1, m]$, D_i is a dimension table schema sch (D_i) = $\{a_i^1, a_i^2, \dots\}$, where $\{a_i^1, a_i^2, \dots\}$ is the set of attributes of D_i. We assume that a_i^1 is the lowest level of D_i so a_i^1 is the primary key of the table D_i.

-For $\forall j \in [1, n]$, F_j is a fact table schema sch (F_j) = $\{a_1^1, \dots, a_m^1, m_j^1, \dots, m_j^w\}$ where $\{m_j^1, \dots, m_j^w\}$ is a set of measurements that can be aggregated F_j according aggregation function [14].

Modeling the user profile

The user profile consists of a set of predicates reflecting the interests of the user in the content of MDB in a given analysis. It is defined by the set of pairs $M_i = (P_i, cp_i)$ where M_i represents the association of preference P_i to its analysis context cp_i. So, the attributes of the table of profiles are the predicate (pred), the degree of interest of the user to the predicate (θ) and context analysis (cp). In Figure 1, we give an example of user profiles.

Num	Ref_user	degree	predicate	contexte_of_preference
1	700010	1	nature='science computer'	ALL
2	800202	0.9	author='Sadok Zerelli'	nature='economic'
3	700011	1	nature='economic'	ALL
4	700010	0.7	author='James Rumbaugh et al'	nature='science computer'
5	800202	0.6	author='Microsoft Press'	nature='managment'
6	700010	0.7	title='Economie du développement'	nature='economic'
7	800202	0.7	title='Economie du développement'	nature='economic'

Figure 1. Set of user profiles.

Definition of a preference

Given a MDB, a preference is defined by $P^A = (pred^A, \theta)$, where :

- A denotes an attribute dimension or measure, possibly associated with an aggregation function.
- $pred^A$ is a disjunction of predicates in the form A op a_i that specifies a condition on the values of a_i A. We assume op ∈ {=, <,>, ≤, ≥, ≠} for numeric attributes and op ∈ {=, ≠} for other types of data.
- Θ is a real number between 0 and 1 indicating the degree of interest of the user to data which is generated by $pred^A$. [14]

Two types of preferences can be distinguished: One absolute, with a degree of interest equal to 1 and independent of context, it can be represented as $M_1 = (P_1, ALL)$ where $P_1 = (pred_1, 1)$. The other is contextual, associated with a context analysis given in the form $M_2 = (P_2, cp_2)$ where $P_2 = (pred_2, \theta_2)$ as the value of $\theta_2 < 1$ and indicates its scope of application.

In the following, we give a general idea of the reformulation of user preferences. For example, if a user expressed this preference: "I am interested only in sales made in Paris." This preference is an absolute preference because it applies to any context. Their reformulation can be represented by:

• IF (user) THEN (city = Paris) or

• P1: (country = 'Paris', degree = 1); M1 (P1, ALL)

Therefore, each preference is reformulated by a predicate that will be included in the initial query if its constraint is verified. The predicates contain only strings of characters and real numbers that represent the degrees of user interest's preferences (degrees ≤ 1). And since only predicates that describe the user profiles are stored in the database, so, we can conclude that the cost of storage of profiles is low and doesn't present a problem of memory.

Define the context of preference

The context of analyzing a preference is defined by : cp = ([F_i (.m^{Fi}/pred/ ... / pred) *], [D_1 ($perd_1$ / ... / pred) *], ..., [D_q ($perd_1$ / ... / pred) *]), where F is the fact analyzed by measuring m^{Fi}, D_1, ..., D_q are the dimensions of analysis, pred is a selection of the attribute or measure x . The sign "*" denotes the absence of a component or the presence of several. Indeed, some elements of cp may be empty. This resulted in the assignment of value ALL to properties. [14]

Valid preference

Preference stored in the profile table is called valid if it relates to the current user context and having the same context analysis or more general than the current analysis (cp_i included or equal to CAC). All preferences are all valid candidates' preferences.

Conflict management

The conflict between the two predicates preferences is an offline managed proactively during the acquisition phase information and updates. The decision to conserve or reject of preferences is at the discretion of the decision maker.

Log file structure

The log file is composed of a set of sessions already posed by the current or the other users. We define the structure of this file par by the set of sessions where each one is posed by one user: ([System Date] [server] [user] [base used] [all requests])*. The symbol '*' denotes the absence of a session or the presence of several.
In Figure 2, we give the structure of the sessions stored in a log file.

```
-- 18/04/2012 18:53 VALR: Execution trace: root@localhost (biblio) Thread ID 2: Commit on Thread ID 2
-- 18/04/2012 18:53 VALR: Execution trace: root@localhost (biblio) Thread ID 2: Executed on Thread ID 2:
select auteur, editeur, ref_ouvrage, ISBN from ouvrage ;
select auteur, editeur, ref_ouvrage, ISBN from ouvrage where nature='commerce'
select auteur, editeur, ref_ouvrage, ISBN, nature from ouvrage where nature='informatique';
-- 18/04/2012 18:54 VALR: Execution trace: root@localhost (biblio) Thread ID 2: Commit on Thread ID 2
-- 18/04/2012 18:54 VALR: Execution trace: root@localhost (biblio) Thread ID 2: Executed on Thread ID 2:
select auteur, editeur, ref_ouvrage, ISBN from ouvrage ;
select auteur, editeur, ref_ouvrage, ISBN,nature from ouvrage where nature='informatique';
-- 18/04/2012 18:54 VALR: Execution trace: root@localhost (biblio) Thread ID 2: Commit on Thread ID 2
-- 18/04/2012 18:55 VALR: Execution trace: root@localhost (biblio) Thread ID 2: Executed on Thread ID 2:
select auteur, editeur, ref_ouvrage, ISBN from ouvrage where nature='commerce'
select auteur, editeur, ref_ouvrage, ISBN,nature from ouvrage;
select auteur, editeur, ref_ouvrage, ISBN from ouvrage where nature='commerce'
select auteur, editeur, ref_ouvrage, ISBN,nature from ouvrage;
```

Figure 2. Example of sessions from the log file

IV. PERSONALIZATION AND RCOMMENDATION

From the current query, user profile and log file, we propose to personalize the current query and recommend to the current user queries from the log file to help him formulate his queries. Figure 3 summarizes the objectives of our approach.

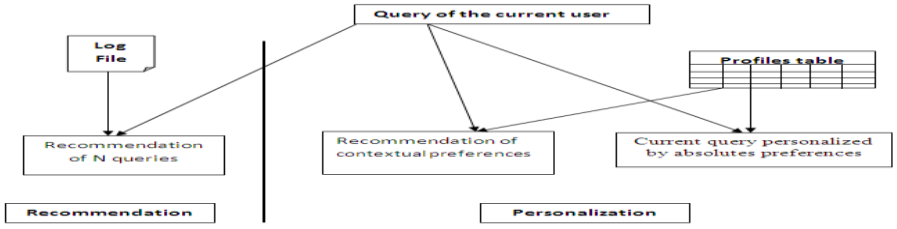

Figure 3. The objectives of our approach

IV.1 PERSONALIZATION

The purpose of personalizing the content of MDBs is to refine the size of data returned by a query posed by the current user. This results in the individualization of DB ignoring the part of the base that the user does not interest and which doesn't present his own needs or interests.

Generally preference that has the most important degree of interest has the highest priority. Therefore, an absolute preference that involves any context is considered higher priority than contextual preference with specific context framework for its implementation. Then, the questions that arise are : Since absolute preference has a higher priority, why do we assume preferences among candidates? Why do we not give more priority to

this type of preferences? Why do we not restrict customization queries at this level? Why do we not remember the user's preferences that he can choose?

In addition, the number of preferences included in the initial query is set by the user or by a given constraint [14]. In this case, some preferences will not be considered and will be ignored although they can meet the needs of users. Until the absolute number of preferences increases, the result size will be reduced while ensuring that data ignored by the increased query are useless to the current user, we proposed to integrate all absolute preferences in the initial query and recommend the K contextual preferences to current user. As a result, our contribution interesting personalization includes two steps. In the first step, we will personalize the current query integrating all absolute preference of the user. A valid search preference is performed as follows: for each record in the table of profiles, we check if a degree of interest of the user is equal to 1 (ɵ = 1) for any context (cp = ALL) if it is the case, we add the predicate to all absolutes preferences.

The second step of personalization is to recommend a set of contextual preferences to the user to remind and help him to formulate his queries. We will rely on [14] to recommend contextual preferences to the current user. Therefore, the construction of custom content is to look for a query R, the set of preferences P = (P^{abs}, P^{rec}), where P^{abs} is a set of absolute preferences that will be integrated directly into the initial query and P^{rec} is a set of contextual preferences that will be recommended to the user. Note that if the set P is empty, the content of the MDB is content that is perfectly adapted to the preferences of the current user, or the user hasn't valid preferences (absolute and contextual).

Thus, the reformulation of the initial query R generate an increased query R' (such that R included or equal to R'). This consists of inserting in the WHERE clause selected predicates in conjunction with those of R to restrict the instances.

IV.2 RECOMMENDATION

Recommendation can be made based on user's preferences to recommend custom queries [8] or simply to recommend preferences. Also, the recommendation may be made based on the sessions recorded in the log file. In addition we note that the method of personalization proposed [13, 14] can be seen as a collaborative method for recommending queries to the current user.

Therefore, our proposal is to recommend queries whose indexes are (p+1) in log file and verifying the following constraint of correspondence: All arcs (arc_i) of the tree of the current query (A^{cac}) must be integrated into the tree of the query (S_p) that its index is p in log file. In addition, we took advantage of personalization, allowing the user select the maximum number of queries recommended (N) and specify the type of query (initial or custom) to make recommendation.

Thus, the proposed recommendation algorithm takes as input parameters S, K and A^{cac} where S is the set of queries log file, K is the number of queries to recommend and A^{cac} represents the tree of specified query (initial or custom).

We describe in the following the proposed algorithm for recommendation queries.

Recommendation Algorithm Input

S : set of queries in log file
A^{cac} : tree of the current query
N : maximum number of queries to recommend
arc_i : attribute or value of current query
Output
R : set of queries to recommend
Begin
$R \leftarrow \emptyset$; $i \leftarrow 0$
While (S** not empty **and** i < N) **do
included ← *true*
For each arc_i of A^{cac} **do**
if ($arc_i \notin S_i$) **then**
included ← false
end if
End for
if (included = true) **then**
$R \leftarrow R \cup S_{i+1}$
i ← i + 1
end if

End while
Return(R)
End

V. EXPERIMENTS

The set of experiments presented in this work were performed on an IBM machine equipped with an Intel Dual-Core N500, with a clock frequency of 2.1 GHz and 3GB RAM running on a Windows XP platform. To evaluate the performance of the two proposed algorithms, we implemented a Java prototype above the MySQL. Also, we used the Mondrian OLAP server by taking advantage of the MDB 'Sales' of 'FoodMart' which is installed with OLAP Services [16] and the database of our university library.

V.1. Personalization : Tests and results

During our study, we performed several experiments with different values of K (which K varies between 0 and 5) and 8 users' profiles. Different users belong to different countries and share the same database.

First test.

In this first experimentation, we took the example of a user who is interested only in sales in various cities in France and did not record his preferences in the profile table. This user has expressed the following query: "The amount and quantity of each product sold."

SELECT ([Product]. [All Products], [Store]. [Country]) ON COLUMNS (([Measures].
[Amount], [Measures]. [Q_sold]) ON ROWS
FROM [Sales];

Since the user did not specify the country of sale, the system will return all data stored in the MDB concerning all sales in all cities in different countries.

Second test.

In the second experiment, we took two examples to demonstrate the effectiveness of personalization in both databases.

Example 1: Sale database

Suppose a user who has benefited from the personalization process by recording his preferences in the profile table. In this experiment, we assumed that the user has an absolute preference and two contextual preferences. The absolute preference P3 is represented by : (Country = 'France', degree = 1) for any context M3 (P3, ALL) and the contextual preferences are as follows: P4 (amount> 250000, degree= 0.8) for context cp4 = (ALL, ALL, ALL, ALL, V.amount) and P5 (Q_sold > 25000, degree= 0.7) for the context cp5 = (ALL, ALL, ALL, ALL, Time.year =2008).

Suppose now that the user has posed the same query in the previous example. As a result, the query formulated by the system is :

SELECT ([Product].[AllProducts], [Store].[Country], [Location].[City], [Time]) ON COLUMNS, ([Measures]. [Amount], [Measures]. [Qty]) ON ROWS
FROM [Sales]
WHERE ([Location]. [City]. [France], [Time]. [2012]);

As a result, the system will only return data for sales in France which is its own absolute preference. Also, this user will be benefited by the recommendation of contextual preferences that may represent his future needs (amount> 600; Q_sold> 25000). Note that if the user has selected the value 1 to K, only the contextual preference which has the highest degree of interest will be displayed.

Example 2: Library database

Suppose a student who belongs to computer science that posed the following query and forgot to specify the nature of books:

SELECT title, author, rating
FROM library;

Returning the result of the personalized query, the system integrated the predicate (nature = 'computer science') in the initial query. Consequently, we prove that the custom query returns fewer records than the initial query.

V.2. Recommendation: Tests and results.

In the following, we describe the proposed framework for recommendation queries. Each query launched by a user in data cube is stored in the log file via the OLAP server. So from this file and a current query, we applied the recommendation algorithm using parameters which are defined by the user (the type of query and the number of queries (N) to recommend).

Example

Consider two students A and B who use the data base of our university library. Suppose that student A interrogates the data base by running the following queries:

• **q1 :** "the works that have economic nature and published by 'Sadok Zerelli' ",

• **q2 :** " the economic works that have published by 'Sadok Zerelli' since 2009 ",

Now suppose that student B interrogates the same data base. This user focuses only on economic works published by 'Sadok Zerelli'.

• **q3 :** "the works that have economic nature and published by 'Sadok Zerelli' ".

By applying the proposed recommendation algorithm, the system searches the log file requests that satisfy the constraint correspondence and returns those queries to the current user in the form of recommendations. So the various analyzes above can help this user to reformulate his next queries and improve his navigation in the cube. In our example, the former analysis performed by student A precisely 'q2' may be recommended to student B because the arcs of the tree of the current query of this user correspond to the arcs of the tree of the query 'q2' posed by student A. Therefore, the following query 'q2' can be used as a recommendation (see Figure 4).

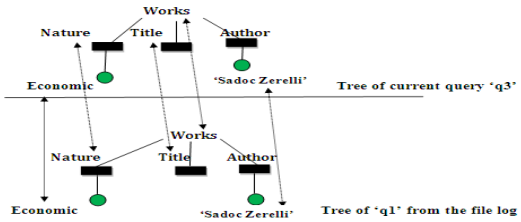

Figure 4. Comparison between the current query 'q3' and the query 'q1'

During the execution of the recommendation algorithm, three cases can be encountered during the verification of the constraint of correspondence between the current query and the query of log file.

• The arcs of two queries are matched, in this case the following query is considered among the queries to recommend.

• The arcs of query log file are included in the set of arcs of the specified query. In this case, the query doesn't interest the user and the system moves to the next query.

• The arcs of the specified query are included in the set of arcs of the query stored in the log file. In this case, this query will be recommended to the current user.

However, the break condition of the loop reached the value of K or arrived at the request of index n (where n is the number of queries in log file).

VI. CONCLUSION AND FUTURE WORKS

In this article, we focused on personalization and recommendation of MDX queries in order to better serve the needs of analysis makers. Our proposals are based on user preferences stored in the profiles table for personalization and queries stored in the log file for the recommendation. To do this, we conducted an experimental study of the proposed algorithms based on the database 'Sale' of 'Foodmart'. We proved experimentally that the algorithm can be used to customize the current user in refining the size of the returned results. In addition, the proposed solution to generate recommendation from log file can help user to formulate and enrich their queries. Prospects for future work include: the definition of new rules for verifying user privileges and the redefinition of rules that express and reformulate user preferences.

REFERENCES

[1]. Aissi S., Gouider M.S., Towards the Next Generation of Data Warehouse Personalization System: A Survey and a Comparative Study, Journal-ref: IJCSI International Journal of Computer Science Issues, Vol 9, Issue 3, No 2, pages 561-568, May 2012.

[2]. Aligon J., Colas S., Marcel P., E. Nègre, Résumés et interrogations de logs de requête OLAP, Extraction et gestion des connaissances (EGC'2011), 2011.

[3]. Bellatreche L., Giacometti A., Marcel P., Mouloudi H., Laurent D., A Personalization Framework for OLAP Queries, In 8th International Workshop on Data Warehousing and OLAP, DOLAP'05, pp. 9–18, 2005.

[4]. Bellatreche L., Giacometti A., Marcel P., Mouloudi H., Personalizationation of MDX queries, Bases de Données Avancées (BDA'06), 2006.

[5]. Bentayeb F., Favre C., URoK : Rok : Roll-up with the k-means clustering method for recommending olap queries, DEXA, pp. 501–515, 2009.

[6]. Colas S., Marcel P., Nègre E., Organisation de log de requêtes OLAP sous forme de site web, In 6èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2010), Djerba, Tunisie, RNTI, Vol. B-6, Cépaduès, Toulouse, 81-95, Juin 2010.

[7]. Favre C., Bentayeb F., Boussaid O., Evolution et personnalisation des analyses dans les entrepôts de données: une approche centrée sur l'utilisateur, XXVème congrès Informatique des organisations et systèmes d'information et de décision (INFORSID 07), pp.308 -323, Mai 2007.

[8]. Giacometti A., Marcel P., Nègre E., A Framework for Recommending OLAP Queries, In 11th ACM International Workshop on Data Warehousing and OLAP , DOLAP'08, pp. 73–80, 2008.
[9]. Giacometti A., Marcel P., Nègre E., Recommending Multidimensional Queries, 11th International Conference on Data Warehousing and Knowledge Discovery DAWAK'09, pp.453-466, 2009.
[10]. Giacometti, A., P. Marcel, E. Nègre, and A. Soulet, Query recommendations for olap discovery driven analysis, In DOLAP, pp.81–88, 2009b.
[11]. Garrigos I., Pardillo J., Mazon J., Trijullo J., A conceptual modeling approach for OLAP Personalization, 28th Conference International Conference on Conceptual Modeling, ER'09, pp. 401-414, November 9-12, 2009.
[12]. Inmon W. H., Building the Data Warehouse, John Wiley and Son (New York, NY, second edition, ISBN : 04771-14161-5, 1996).
[13]. Jerbi H., Personnalisation d'analyses décisionnelles sur des données multidimensionnelles, 2012, École doctorale Mathématiques, Informatique et Télécommunications (Toulouse, Haute-Garonne), en partenariat avec Institut de Recherche en Informatique de Toulouse (équipe de recherche), à Toulouse 1, available at : http://www.theses.fr/2012TOU10009.
[14]. Jerbi H., Ravat F., O. Teste, G. Zurfluh, Personnalisation du contenu des bases de données multidimensionnelles", 6èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2010), Djerba, Tunisie, RNTI, Vol. B-6, Cépaduès, Toulouse, 5-19, Juin 2010.
[15]. Nègre E., Quand la recommandation rencontre la personnalisation. Ou comment générer des recommandations (requêtes MDX) en adéquation avec les préférences de l'utilisateur, Technique et Science Informatiques 30(8): 933-952, 2011.
[16]. Pentaho Corporation: Mondrian open source OLAP engine (2009), Available at http://mondrian.pentaho.org.