An Improved Multidimensional Query Language for Oil and Gas Development Data Warehouse

Jidong Zhao, Tingzhen Yao
School of Computer Science, Xi’an Shiyou University, Xi’an 710065, China
xazjd@126.com

Abstract. Multidimensional data analysis is the most central technique for processing data in data warehouse and On-Line Analysis Processing (OLAP). In order to solve the problem of multi-dimensional analysis expression and query processing in oil and gas development data warehouse, this paper proposes an improved multi-dimensional query language application method to realize effective interaction between query terminal and OLAP server. This formal language has the following advantages, such as simplicity, flexibility and easy to use. It also meets the needs of multi-dimensional query for oil and gas data analysis, and has various display modes. The results can be displayed in two-dimensional space. And it provides an extended function beyond the standard MDX, simple grammar format, concise and clear semantic expression, and is conducive to query processing conversion.

1. Introduction
Oil and gas production information data has the characteristics of large amount of data, scattered data, diverse types and complex storage formats. It needs to analyse and query oil and gas production information data from multiple angles, sides and levels in Reservoir engineering analysis method. Data analysis of oil and gas production information by on-line analytical processing technology can provide effective production guidance, discover abnormal production situation in time, ensure normal, safe and economical operation of oil and gas fields, and achieve the goal of improving the efficiency of oil and gas field development[1-2].

Thus, the research on multi-dimensional data analysis method and formal language of oil and gas production information can effectively improve the efficiency of subject-oriented information acquisition, greatly help oil and gas production decision makers to adjust strategies, reduce risks and make correct decisions.

2. Requirements for Query Language in Oil and Gas Development Data Warehouse

2.1. Multidimensional Data Analysis to Satisfy Industry Demand
Oil and gas development data warehouse provides a multi-dimensional view of data cube. Users can observe data in data warehouse from multiple angles by means of multi-dimensional query language. It is convenient and fast to realize slicing, roll-up, drilling and rotation analysis, and supports multi-dimensional display of analysis results.

In the data warehouse system for oil and gas development, the terminal equipments for multidimensional analysis are not only desktop computers and mobile computers, but also some other portable devices, such as small industrial dedicated computers based on PC industrial standards, and other query terminals equipped with streamlined operating systems. This requires the multi-
dimensional query expressions used on terminal devices should have lower complexity, good expression ability and basic functions of multi-dimensional analysis. Furthermore, it should be easily installed and used without excessive restrictions on operating system environment. The OLAP analysis service architecture is shown in Fig.1.

![Fig.1 Oil and Gas Development Data Warehouse OLAP Analysis Service Architecture](image)

2.2. Multidimensional Query Language
The multidimensional expression MDX is provided in SQL Server Analysis Services, but the language is overcomplicated and presents limitations on certain query requirements. Therefore, this paper designs a special multi-dimensional query analysis language according to the characteristics of oil and gas development data warehouse, which is based on the standard MDX compiler environment.

Analysis Services provides an extensible architecture. When building a business intelligence application project, developers can realize some special functions of the business application system by writing a set of user-defined functions (these user-defined functions provide services on Analysis Services and can be invoked and executed by MDX) [3].

To sum up, this paper extends MDX and realizes the multi-dimensional analysis formal language Oil and Gas Multi-Dimension Expression (OG_MDX) based on the multi-dimensional conceptual model of oil and gas development data warehouse with the help of specified services provided by user-defined functions on the server side.

3. Design of OG_MDX

3.1. Design Principles of OG_MDX Language
The data warehouse system for oil and gas development has the characteristics of wide coverage area, multiple types of supporting software, high and low hardware performance, and complex educational structure of users. In the design of multi-dimensional query language, it is necessary to consider the application scenarios of oil and gas data management and the hardware and software environment, while taking into account both the expressive ability and better usability.

There are five basic principles to follow when designing the improved multidimensional query language OG_MDX:

First, it should meet the functional requirements of multidimensional analysis and processing of oil and gas development data warehouses, focusing on application services;
Second, it should reduce the complex redundancy function of MDX, keep it simple and easy to use, and the language function only needs to meet the multi-dimensional query requirements of the oil and gas development data warehouse;

Third, it should use high-level language programming interface component expansion for some special analysis functions;

Fourth, it should be used with Excel VBA programming for multidimensional dynamic results display;

Fifth, the language form should be standardized and convenient for interpretation module design to realize the interaction between OG_MDX and OLAP server.

3.2. Language Description of OG_MDX
In the process of designing OG_MDX, a large number of basic grammatical functions are extracted from the grammar set of MDX language, which is the basis of realizing OG_MDX language. This can be consistent with the user interface of the mainstream data warehouse system. For some complex grammatical functions, it will be extended in the future research. Considering the characteristics of data cube in oil and gas development data warehouse system, some special functions are added in OG_MDX language.

The basic structure of OG_MDX grammar is as follows:

```
SELECT <axis_specification_col>[,<axis_specification_row>]
FROM <cube_name>
[WHERE <slice_specification>,<slice_specification>…];
```

Among them, `<axis_specification_col>` and `<axis_specification_row>` represent axis dimensions in the form of:

```
< axis_specification_col > ::= <set> [<dim_props>] ON (dimension_name)
< axis_specification_row > ::= <set> [<dim_props>] ON (dimension_name)
```

Simplified forms can also be used:

```
< axis_specification_col > ::= <set> [<dimension_expression >]
< axis_specification_row > ::= <set> [<dimension_expression >]
```

`<slice_specification>` represents the slice dimension and is used to restrict the returned data to specific dimensions and membership conditions. `<cube_name>` represents the data cube name.

The data cube Wells is assumed to be a four-dimensional data cube with unit dimension (Factory), well name dimension (Well), block dimension (Section) and time dimension (Time) [4]. A hierarchical relationship is defined on each dimension, and the hierarchical structure is shown in Fig.2.

The multi-dimensional query statements based on OG_MDX are as follows:

```
SELECT
{[Year].[2016].[Q3],[Year].[2017].[Q3]} ON (Time),
{([Section].[ Block 1],[Corp].[Oil Production Company 1]),
([Section].[ Block 1],[Corp].[Oil Production Company 2])
```

Fig.2 Dimensional Hierarchy Diagram of Wells Data Cube
For example, in the third quarter of 2016 and 2017, we inquired about the oil production of each company (Oil Production Company 1, Oil Production Company 2 and Oil Production Company 3) in Block 1 and Block 3, respectively. It is required that the vertical coordinates should be quarterly time and the abscissa should be blocks and oil production companies. The multi-dimensional query statements based on OG_MDX are as follows:

```
SELECT
```
3.3. Improvement of SELECT clause and WHERE clause

The SELECT clause draws on the MDX basic structure and syntax to select the dimensions and members to return, consisting of the SELECT keyword and multiple axis dimensions. The axis dimension is described by the set of tuples, the keyword ON, and the axis name. In order to achieve the intended design goals, three improvements have been made to the MDX SELECT clause:

(1) For the display of multidimensional results in a two-dimensional space, OG_MDX specifies that the SELECT clause can only have a maximum of two axis directions.

(2) Use the data cube feature name as the axis name after the ON keyword. The axis dimension is composed of isomorphic tuples, and the dimension names of the tuples in each direction constitute a tuple structure feature. It can also be expressed as a standard dimension member. In which case, the keyword ON and the axis name are omitted.

(3) The format of the members is defined by the name of the hierarchy and the value of the attribute. The semantic structure is [hierarchical name].[attribute value]([attribute value]......).

The WHERE clause limits the returned data to specific dimensions and membership conditions. In order to make more flexible restrictions on the cube, the WHERE clause has also been grammatically modified from three aspects:

(1) The slice dimension of OG_MDX is designed to be relatively simple. A WHERE clause consists of multiple slice dimensions. The structure of a slice dimension is the same as the axis dimension, but can only represent information on one dimension. The tuple of such a sliced dimension will consist of a single member, thus syntactically treating the sliced dimension as a collection of members.

(2) Define attribute expressions in the slice dimension to limit non-hierarchical attributes. In this way, in the slice analysis of OG_MDX, not only the hierarchical attributes can be restricted, but also the non-hierarchical attributes can be restricted.

(3) To display the aggregated results specified by the user in the crosstab cell, use a measure dimension as the slice dimension to restrict.

After improving the MDX language, the basic grammar of OG_MDX is formed. It saves the development cost while ensuring the function and usability of the language. The improvement process follows the design principles, and the language function achieves the expected design goals.

3.4. Further Improvement of OG_MDX Formal Language Using Stored Procedures

Stored procedures can be written in a high-level programming language, allowing developers to write common code at once, storing code in a specific location, and reusing stored common code in other stored procedures, operational expressions and user queries, thus simplifying the development and implementation of the database [5].

In order to ensure the security of oil and gas development data warehouse system, all OLAP operations are first considered to be executed on the server side. Using the high-level language environment to write stored procedures, we can place the stored procedures to achieve specific multi-dimensional analysis operations on the server side, and call them on the server side for execution, which can achieve the desired design goals of the system.

The oil and gas development data warehouse mainly adopts the following three methods to expand the function of OG_MDX.

(1) Use the .NET user-defined function to improve MDX.
(2) Use the AMO stored procedure to improve MDX.
(3) Use the AMOMD Server stored procedure to improve MDX.
4. Design and Implementation of OG_MDX Interpretation Module

The interpretation module is between the front-end query analysis tool and the OLAP server. It is the interface between OG_MDX and Analysis Services. The main functions are as follows:

1. Check the lexical, grammatical and semantics of the OG_MDX query statement;
2. Convert the OG_MDX query statement into a standard multidimensional expression according to the syntax rules of OG_MDX;
3. Manage various types of metadata involved in the multidimensional analysis process;
4. Handling all errors arising during the interpretation process.

The OG_MDX interpretation module architecture is shown in Fig. 3.

The lexical analysis sub-module scans the OG_MDX statement from left to right, identifies word symbols (keywords, identifiers, numbers, operators, special symbols, etc.) according to lexical rules, and generates a terminator stream for parsing; filters out the explanations and blanks in OG_MDX statements (caused by spaces, tabs and carriage returns); records the position information of the input OG_MDX string. When a lexical or grammatical error occurs, the error handling sub-module can report the exact location of the input OG_MDX statement error.

The syntax analysis sub-module reads the terminator stream from the lexical analysis sub-module, and recognizes all kinds of grammatical components from the terminator stream. According to the designed OG_MDX grammar, the structure of the OG_MDX statement can be found and the grammatical errors in the OG_MDX statement can be detected. If grammatical errors are detected, the error processing sub-module is called for processing; if the grammar is correct, the grammar tree and symbol table of OG_MDX statement are constructed according to the sequence of actions generated by grammatical analysis, which are provided for the use of the semantic processing sub-module.

The Semantic processing sub-module checks the correctness of the syntax of OG_MDX query statement and confirms that the statement is meaningful.

![OG_MDX Interpretation Module](image)

**Fig. 3 OG_MDX Interpretation Module**

Interpretation and transformation sub-module is the core component of this module. When the OG_MDX query statement is checked lexically, grammatically and semantically, the OG_MDX query statement string will be converted to the standard MDX query symbol string according to the OG_MDX grammatical rules. The standard output multi-dimensional expression contains only the MDX statement which used in the specific query and analysis function. Other irrelevant symbols have been filtered.
5. Conclusion
This paper discusses the advantages and disadvantages of MDX application in oil and gas development data warehouse, analyses the current situation and characteristics of oil and gas exploration and development data management, and puts forward an improved multi-dimensional analysis formal language OG_MDX which is applied to oil and gas development data warehouse system. This multi-dimensional query language is based on MDX standard grammar and simplified and optimized according to the background of oil and gas development project. Through the user-defined function (stored procedure) technology, OG_MDX language is further extended to make its multi-dimensional analysis function more powerful and easy to implement in the data warehouse for oil and gas development. In practical applications, OG_MDX not only meets the query requirements of conventional computer terminals, but also responds quickly to the multi-dimensional analysis needs of mobile terminal.

Acknowledgments
This work was financially supported by the National Natural Science Foundation of China (41301480).

References
[1] Diamantini Claudia, Potena Domenico, et al. Multidimensional query reformulation with measure decomposition[J]. Information Systems, 2018, 78(2):23-39.
[2] Meisen Philipp, Keng Diane, et al. TIDAQL: A query language enabling on-line analytical processing of time interval data[C]. 17th International Conference on Enterprise Information Systems, Proceedings, 2015:54-66.
[3] Da Silva, Paulo Caetano, et al. LMDQL: Link-based and multidimensional query language[C]. International Conference on Information and Knowledge Management, 2009:89-96.
[4] Mao Keji, Zhao Xiaomin, et al. Data Storage Scheme Supporting for Multidimensional Query[J]. International Journal of Distributed Sensor Networks, 2013, 2013(3):140-154.
[5] Ye Xuefang, He Yue. Designing multi-dimensional analysis in oil wells information system[J]. Computer Applications and Software. 2010, 27(11):16-19 (in Chinese).