MODELING OF AN ANALYTICAL DATABASE SYSTEM

Alex Sandro Romeu de Souza Poletto and Jorge Rady Almeida Junior
Fundação Educacional do Município de Assis (FEMA) – Av..Getúlio Vargas, 1200. cep:19807-634 – Assis - SP - Brazil
Escola Politécnica-Universidade de São Paulo – Av. Prof. Luciano Gualberto, 158. cep:05508-900 - São Paulo - Brazil

Keywords: Analytical database, trigger, procedure, operational database, decision taking, SQL.

Abstract: This paper describes a modeling for constructing an analytical database, with the objective to store historical values and also the most recent values starting from operational databases. The first function of this modeling is to map operational database into analytical database, using their E-R diagrams. For that, we created ten steps, which will support the mapping. The second function is to make available mechanisms for generation, transport and storage of these historic data. For that, we specified triggers and procedures for each step.

1 INTRODUCTION

This work proposes a new modeling form for the Analytical Databases. The importance attributed to the data modelling techniques, considering the several data transformation and transference phases between the diverse system types.

In recent years, the need for storing and recovering historical data has presented great growth. The main reason for this is because operational databases have not been designed to attend to such necessities, nowadays so important for the most of applications. Usually, the purpose of operational databases is to store only the most recent value. The need to have previous data is taking the organizations to look for resources that provide an efficient model and storage of historical data.

For a good administration of an organization, it is important that database system supplies all the data evolution states that is, all the values assumed by the data since its creation. Such evolution can be of great importance for a future process of decision taking.

Usually, to prevent from the excessive growth of the operational databases, many of the past referring values are discarded, that is, they do not become permanent.

For many organizations, the storage of historical data is becoming, a great problem, because there exist few adapted tools for such process.

So, this work has the purpose to make possible the development of mechanisms and techniques that assist in the mapping, storage and use of analytical data, resulting in the Analytical Databases, having as basis the Operational Databases available in the organizations.

Section 2 describes triggers and procedures that are the practical resources that will be used for the generation of the analytical data. Section 3 describes the proposal of an analytical database and on section 4 we make our conclusions about this work.

2 TRIGGERS AND PROCEDURES

Triggers and procedures can also cooperate to the end users application development tools, adding processing resources. Moreover, they also make possible to declare variables based on database attributes, making the code independent from physical changes in the attributes.

Using these resources it is possible to reduce the number of code lines needed by tools used in the development of end users applications. So, many business rules, constraints and integrity rules can be programmed directly in the database, and so, the end application may contain only basic instructions, without the need to guarantee the integrity every time a routine is programmed. It becomes the end applications more agile, with a simpler code, and consequently a better total performance of the application.

Sandro Romeu de Souza Poletto A. and Rady Almeida Junior J. (2007).
MODELING OF AN ANALYTICAL DATABASE SYSTEM.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - DISI*, pages 477-481
DOI: 10.5220/0002362804770481

3 ANALYTICAL DATABASE MODELING

The analytical database has a more complete and rich storage, comparing to the operational environment, and has the specific objective to help the administrators in the medium and long time decision taking process.

It is necessary that the analytical database be more specialized than an operational database, offering historic data for analysis.

When we talk about a more specialized database, we mean a system with different objectives from operational databases, about data storage states. In this case, the specialization has the main objective to supply data for decision taking based on business planning.

The main objective of this work is to propose and describe a modeling and implementation approach for an analytical database, as shown on Figure 1.

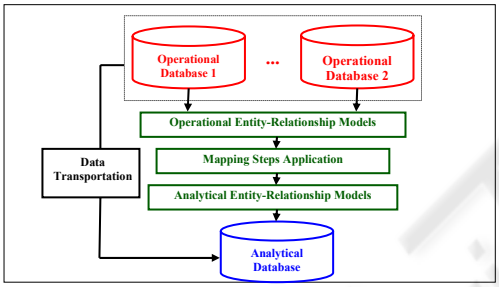

Figure 1: General diagram of the work proposal.

The adopted way of storage is using only one repository, that is, all the operational databases must originate only one analytical database, which will receive from the operational database all the data necessary for the decision taking process.

Such task is divided into three stages, as shown in Figure 2 and discussed below. It is important to say that the three Stages proposed here must be based on the business planning for the organization. This way, it is essential the participation of business analysts, as well as the system analysts in this proposal.

3.1 Stage A: Mapping E-R Diagrams into Unified E-R Diagram

This stage intends to map the several E-R diagrams used for the operational database modeling into only one unified E-R diagram, with the objective to join, distinguish and standardize all the entities sets.

With this stage, we intend to reduce some problems, like: same data with different names, different data with same name, different measurement unities, null values, duplicate keys, and so on. These problems may occur especially because it is possible to work with more than one operational database.

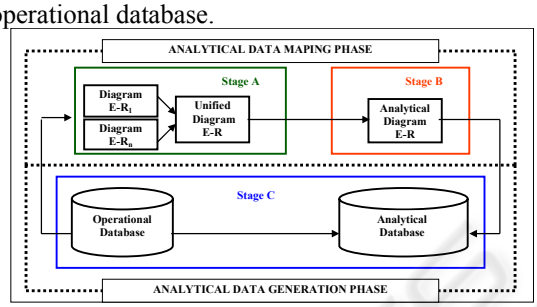

Figure 2: Detailed diagram of the work proposal.

Step A1: Identification of entities sets, which store distinct data with equal names.

There can be several operational databases, which will compose the analytical database. In such databases, there can be sets of entities which have same names, but storing distinct data types. As it is not possible to work this way, it is necessary to identify such entities sets and give them different denominations.

Step A2: Identification of entities sets that store common data with distinct names.

There can be several operational databases, which will compose the analytical database. In such databases, there can be entities sets which have different names, but storing common data types. So, it is necessary to identify such entities sets and, eventually, give them the same denominations.

Step A3: Identification of attributes that store common data with distinct names.

There can be several operational databases, which will compose the analytical database. In such databases, there can be, in the several entities sets identified in the Step A2, attributes with different names, but storing common data.

Step A4: Identification of primary key attributes – value duplication

It is possible the occurrence of duplication of attribute values at the moment of the analytical data generation, because it is made the unification of entities sets, as described in the Steps A2 and A3. Such duplication can occur even with attributes used as primary keys, what is not admissible. So, this step seeks to solve this kind of problem.

Step A5: Identification of foreign key attributes.

The main reason of this step is due to primary key values change, as suggested at Step A4, and as

these values could be used as foreign key by other entities sets, allowing the integrity lost. Then, these new values must be propagated to every entities sets, to avoiding the integrity lost among data.

3.2 Stage B: Mapping the Unified E-R Diagram into Analytical E-R Diagram

This stage has as its objective to map the unified E-R diagram, obtained at the end of Stage A, into an analytical E-R diagram, supporting the analytical database with a structure that allows the storage of historical data.

The main objectives of this stage are: reduction of the operational entities and attributes data structures; inclusion of time attributes; addition of derived attributes; modification of data relationship into entities sets and the accommodation of different granularity levels.

Step B1: Selection of entities and attributes sets, which will be stored in the analytical database.

It is essential to select all the entities and attributes sets, which will really be in the process of the analytical database generation. At this step it will also be possible to check which entities sets are purely operational, because this kind of data is not included in the analytical database generation process.

Step B2: Inclusion of time element in the structure of entities sets.

It is very important, at several cases, the inclusion of attributes that will control the period of time when the entities values will be valid. That is, at a certain time, such value was considered the most recent, or valid. For this reason, it is necessary the inclusion of time attributes.

At some cases, it may be necessary to add a new set of entities to store all the occurrences of value changes. This step is one of the most important of this proposal.

Step B3: Inclusion of derived attributes.

At some cases, it may be necessary to take decisions at a short space of time. It may be necessary to make very complex queries, which need to access a great volume of tuples, and also to make many calculations.

So, in this step it is made the addition of attributes derived from other ones as, for example, the amount of a selling, the age of a person, etc.

Step B4: Creation of entities sets

In some cases, it may be necessary to store multivalued attributes, that is, attributes that need to store many values for the same tuple.

Step B5: Application of the different granularity levels

This step is important when we consider the performance factor. If we have a great level of data summarization, we will have also a lower volume of data storage

There are many ways to increase the granularity: by summary, by average, by limit values, or by and weekly or monthly values (Kimball, 2002).

3.3 Stage C: Generation of the Analytical Data

This stage has the objective to generate, transport and store the analytical data, originated from the operational databases to the analytical database, using triggers and procedures.

All this generation, transport and storing process must be started based on the diagrams generated by Stage B. Figure 3 shows how this stage must occur.

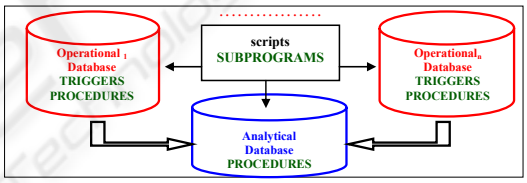

Figure 3: General Schema of Stage C.

The structures that will compose the analytical database are based on analytical E-R diagrams.

Using the operations made by the end users, which will automatically call the triggers and procedures programmed directly at operational database, data from operational databases will be immediately and remotely transported to the analytical database every time that occurs some transactions, which affects the tuples and/or the attributes selected for the analytical data generation.

The next items describe the subprograms, that are, the generic triggers and procedures necessary to complement this proposal.

Step C1: Generation of Trigger for Step **B1**.

The objective of this step is the creation of a trigger to implement Step B1. The generic trigger proposed for this step is specified at Figure 4.

```
CREATE OR REPLACE TRIGGER T_STEP_C1
BEFORE/AFTER INSERT OR UPDATE ON BDO_Table
FOR EACH ROW
BEGIN
  IF INSERTING THEN
    INSERT INTO BDA_Table (BDA_Key_1, BDA_key_2, . . ., BDA_Attribute_1, BDA_Attribute_2, . . . .)
    VALUES(:NEW.BDO_Key_1, NEW.BDO_Key_2, . . .,
    :NEW.BDO_Attribute_1, :NEW.BDO_Attribute_2 . . . .);
  ELSE
    IF :NEW.BDO_Attribute_1 <> :OLD.BDO_Attribute_1 OR
            NEW.BDO_Attribute_2 <> :OLD.BDO_Attribute_2 OR . . . .
      INSERT INTO BDA_Table (BDA_Key_1, BDA_Key_2, . . ., BDA_Attribute_1, BDA_Attribute_2, . . . .)
      VALUES(:OLD.BDO_Key_1, :OLD.BDO_Key_2, . . . .,
      :NEW/OLD.BDO_Attribute_1, :NEW/OLD.BDO_Attribute_2, . . . .);
  END IF;
END.
```

Figure 4: Specification of Step C1.

Step C2: Generation of Trigger for Step **B2**.

This step complements Step C1 and has the objective to include dates in the tuples that have changes in attributes values. The generic trigger proposed for this step is specified at Figure 5.

```
CREATE OR REPLACE TRIGGER T_STEP_C2_1
BEFORE/AFTER INSERT OR UPDATE OF BDO_Attribute_X ON BDO_Table
FOR EACH ROW
BEGIN
  IF INSERTING THEN
      INSERT INTO BDA_Table (BDA_Key_1, BDA_Key_2, . . . . .,
      BDA_Attribute_1, BDA_Attribute_2, . . ., BDA_Attribute_Time_1)
      VALUES(:NEW.BDO_Key_1, NEW.BDO_Key_2, . . . .,
      :NEW.BDO_Attribute_1, :NEW.BDO_Attribute_2 . . . . ., Date_Day);
  ELSE
    UPDATE BDA_Table SET BDA_Attribute_Time_2=Date_Day
    WHERE BDA_Table.BDA_Attribute_X=:OLD. BDA_Attribute_X AND
    BDA_Table.BDA_Key_1=:OLD.BDO_Key_1 AND BDA_Table.BDA_Key_2=:OLD.BDO_Key_2 AND . . .
    AND BDA_Table.BDA_Attribute_Time_2 is NULL;
    INSERT INTO BDA_Table (BDA_Key_1, BDA_Key_2, . . . .,
     BDA_Attribute_1, BDA_Attribute_2, . . ., BDA_Attribute_Time_1)
     VALUES(:NEW.BDO_Key_1, NEW.BDO_Key_2, . . . .,
     :NEW.BDO_Attribute_1, :NEW.BDO_Attribute_2 . . . .,Date_Day);
  END IF;
END;
```

Figure 5: Specification of Step C2.

Step C3: Generation of Trigger for Step **B3**

The objective of this step is the creation of a trigger to implement Step B3. The generic trigger proposed for this step is specified at Figure 6.

```
CREATE OR REPLACE TRIGGER T_STEP_C3
BEFORE/AFTER INSERT ON BDO_Table
FOR EACH ROW
BEGIN
  INSERT INTO BDA_Table (BDA_Key_1, BDA_Key_2, . . . . .,
  BDA_Attribute_1, BDA_Attribute_2, . . . ., BDA_Attribute_Total_1, . . . . ,
  VALUES(:NEW.BDO_Key_1, NEW.BDO_Key_2, . . . ., :NEW.BDO_Attribute_1, :NEW.BDO_Attribute_2, . .
  . . ., :NEW.BDO_Attribute_Calc_1*:NEW.BDO_Attribute_Calc_2, . . . .);
END.
```

Figure 6: Specification of Step C3.

Step C4: Generation of Trigger for the Step **B4**

The objective of this step is the creation of a trigger to implement Step B4. The generic trigger proposed for this step is specified at Figure 7.

```
CREATE OR REPLACE TRIGGER T_STEP_C4
BEFORE/AFTER INSERT OR UPDATE OF BDO_Attribute_1, BDO_Attribute_2, BDO_Attribute_3, . . . .  ON
    BDO_Table
FOR EACH ROW
BEGIN
  IF INSERTING THEN
      INSERT INTO BDA_Table(BDA_Key_1, BDA_Key_2, . . . . .,
      BDA_Attribute_1,  BDA_Attribute_2, BDA_Attribute_3, . . . . , BDA_Attribute_Time_1)
      VALUES(:NEW.BDO_Key_1, :NEW.BDO_Key_2, . . . .,
      :NEW.BDO_Attribute_1, :NEW.BDO_Attribute_2, :NEW.BDO_Attribute_3, . . . . ,Date_Day);
  ELSE
      UPDATE BDA_Table SET BDA_Attribute_Time_2=Date_Day
      WHERE BDA_Table.BDA_Key_1=:OLD.BDO_Key_1 AND
      BDA_Table.BDA_Key_2=:OLD.BDO_Key_2, . . . , AND BDA_Table.BDA_Attribute_Time_2 is NULL;
      INSERT INTO BDA_Table(BDA_Key_1, BDA_Key_2, . . . . .,
      BDA_Attribute_1,  BDA_Attribute_2, BDA_Attribute_3, . . . . , BDA_Attribute_Time_1)
      VALUES(:NEW.BDO_Key_1, :NEW.BDO_Key_2, . . . .,
      :NEW.BDO_Attribute_1, :NEW.BDO_Attribute_2, :NEW.BDO_Attribute_3, . . . . ,Date_Day);
  END IF;
END;
```

Figure 7: Specification of Step C4.

Step C5: Generation of Trigger for the Step **B5**

The objective of this step is the creation of a procedure to implement Step B5. The generic procedure proposed for this step is specified at Figure 8.

4 CONCLUSIONS

This work described the creation of a historic database, which can be very useful for decision taking process. Besides, there will not be necessary the acquisition of expensive tools to make the generation of the analytical database, because the entire process can be implemented using triggers and procedures.

It must be emphasized the implementation facility and the low cost of the solution proposed in this work, especially when compared to data warehouse technology, which is much more complex and expensive.

For future work, we intend to study a real case, to show the applicability of this solution.

```
CREATE OR REPLACE PROCEDURE P_STEP_C5
 (Variable_Filter_1 IN BDA_Table.BDA_Attribute_1%TYPE,
  Variable_Filter_2 IN BDA_Table.BDA_Attribute_2%TYPE, . . . . )
IS
 Variable_1 BDO_Table.BDO_Attribute_1%TYPE; Variable_2 BDO_Table.BDO_Attribute_2%TYPE;  . . . . .;
 CURSOR BDO_Table_CURSOR IS
   SELECT BDO_Key_1, BDO_Key_2, . . . . , BDO_Attribute_1, BDO_Attribute_2, . . . . ,
   Sum(BDO_Attribute_Calc_1* BDO_Attribute_Calc_2), Sum(BDO_Attribute_Calc_3, . . . .)
   INTO Variable_1,Variable_2, . . . .  FROM BDO_Table
   WHERE BDO_Key_1=Variable_Filter_1 OR/AND . . . .
   OR BDO_Attribute_1=Variable_Filter_2 OR/AND . . . .
   GROUP BY BDO_Key_1, BDO_Key_2, . . . . , BDO_Attribute_1, BDO_Attribute_2, . . . ;
BEGIN
  OPEN BDO_Table_CURSOR;
  LOOP
    FETCH BDO_Table_CURSOR INTO Variable_1, Variable_2, . . . . ;
    EXIT WHEN BDO_Table_CURSOR%NOTFOUND;
    DELETE FROM BDA_Table WHERE BDA_Key_1=Variable_Filter_1 OR/AND . . . .
    OR BDA_Attribute_1=Variable_Filter_2 OR/AND . . . . ;
    INSERT INTO BDA_Table (BDA_Key_1, BDA_Key_2, BDA_Attribute_1,
    BDA_Attribute_2, BDA_Attribute_3, . . . . ) VALUES(Variable_1,Variable_2, . . . . );
  END LOOP;
  COMMIT;
END;
```

Figure 8: Specification of Step C5.

ACKNOLEDGEMENTS

The authors thank the support of FEMA (Fundação Educacional do Município de Assis) and FUSP (Fundação da Universidade de São Paulo), in the development of this work.

REFERENCES

Corey, Michael; Abbey, Michael; Abramson, Ian; Taub, Ben. *Oracle 8i Data Warehouse*. Editora Campus, 2001.

Date, C J.; *Introdução a Sistemas de Banco de Dados*. 8ª edição – Rio de Janeiro: Elsevier, 2003.

Gonçalves, Márcio. *Extração de Dados para Data Warehouse*. Editora: Axcel Bookes do Brasil, 2003.

Inmon, W. H.; Terdeman R. H.; Imhoff, Claudia. *Data Warehousing – Como transformar informações em oportunidades de negócios*.São Paulo: Berkeley, 2001.

Kimball, Ralph. *Data Warehouse Toolkit; o guia completo para modelagem dimensional*. Editora: Campus, 2002.

Kochhar, Neena; Gravina, Ellen; Nathan, Priya. *Introdução ao Oracle: SQL e PL/SQL*. Volume 1 e 2. Editora: Oracle Education, 2000.

Silberschatz, Abraham; Korth, Henry F.; Sudarschan, S. *Sistemas de Bancos de Dados*. 3ª edição - São Paulo. Editora: Makron Books, 1999.

Vassiliadis, Panos; Simitsis, Alkis; Skiadopoulos, Spiros. *Conceptual Modeling for ETL Processes*. National Technical University of Athens. McLean, Virginia, USA, 2002.