International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:11, 2009

Business Rules for Data Warehouse

Rajeev Kaula

***Abstract*—**Business rules and data warehouse are concepts and technologies that impact a wide variety of organizational tasks. In general, each area has evolved independently, impacting application development and decision-making. Generating knowledge from data warehouse is a complex process. This paper outlines an approach to ease import of information and knowledge from a data warehouse star schema through an inference class of business rules. The paper utilizes the Oracle database for illustrating the working of the concepts. The star schema structure and the business rules are stored within a relational database. The approach is explained through a prototype in Oracle's PL/SQL Server Pages.

***Keywords*—**Business Rules, Data warehouse, PL/SQL Server Pages, Relational model, Web Application.

I. Introduction

BUSINESS rules and data warehouse are concepts and technologies that impact a wide variety of organizational tasks. Each area has evolved independently, impacting application development and decision-making.

A data warehouse is developed to support a broad range of organizational (decision) tasks [5]. It is an organized collection of large amounts of structured data, designed and intended to support decision making in organizations. A data warehouse models data differently from transactional databases. There are many models for structuring warehouse data like star schema, snowflake schema, and constellation schema. Of these, the most common model for structuring data is the star schema [11]. The import of information and knowledge from a data warehouse is a complex process that requires understanding of the schema structure and the underlying business environment.

Business rules, on the other hand, are abstractions of the policies and practices of a business organization. Business rules reflect the decisions needed to accomplish business policy and objectives of an organization [3, 4, 6, 12]. Business rules specification enables an organization to better understand its operating environment, along with the ability to assert business structure, control and influence over organizational tasks [12].

Business rules are generally developed as an addendum to transactional database development [3, 4]. They are analyzed from the perspective of either extending an entity relationship model , or protect integrity within a relational database , or assist in the modeling of applications using events to invoke them [1, 6, 12]. Business rules systems are separate systems that assist in the management and execution of business rules pertaining to computational and integrity validation of business logic [6].

Rajeev Kaula is with Computer Information Systems Department, Missouri State University, Springfield, MO 65897 USA (phone: 417-836-5666; e-mail: rajeevkaula@ missouristate.edu).

There are many classes of business rules [6, 7, 8, 9, 13]. One class of business rules referred as inferences facilitates the creation of a new piece of information or knowledge. Such rules typically express decision-making guidelines. These business rules are expressed declaratively in condition-action terminology represented as IF condition THEN action format A condition is some constraint, while the action clause reflects the decision or advice. Figure 1 shows an example of a business rule that describes a set of constraints applicable for approving a loan application.

IF	credit risk is High AND debt to income is less than 40% AND loan requested is less than $50,000
THEN	approve with 5% APR

Fig. 1 Sample Business Rule

Since the declarative nature of business rules are easier to grasp, it is possible to extend the decision support capabilities of business rules beyond transactional databases to data warehouse. Such an application of business rules simplifies the generation and import of information and knowledge from the data warehouse, thereby improving decision support.

This paper outlines an approach to ease import of information and knowledge from a data warehouse star schema through an inference class of business rules. To facilitate such retrieval, the star schema structure and the business rules are stored within a relational database. The approach is illustrated on an Oracle 10g database through a prototype in PL/SQL Server Pages [2, 10]. PL/SQL server pages is a server-side scripting approach for developing database driven dynamic Web pages. The PL/SQL server page uses Oracle's primary database language PL/SQL as a scripting language along with HTML to generate database driven Web pages. The paper now outlines the mechanism for mapping business rules to a star schema, followed by its implementation through a Web prototype.

II. MAP BUSINESS RULES TO STAR SCHEMA

Mapping of business rules to a data warehouse star schema involves (i) understanding of the star schema structure –

dimensions and fact; (ii) representation of star schema instance as a business rule; (iii) nature of business rules representation in database; and (iv) matching of business rules with star schema for rule validation. Each of these features is now explored.

A. Star Schema Structure

The basic premise of a star schema is that information can be classified into two groups: facts and dimensions. Facts are the core data elements one is analyzing, while the dimensions are attributes about the facts. For example, units of items sold are facts, while the criteria to analyze item sale are dimensions.

From a data modeling perspective, a star schema consists of tables referred as dimensions and fact. The fact table is joined to dimension tables using foreign key references. The primary key of the fact table is usually a composite key that is made up of all of its foreign keys. The fact table contains data that represents a measure that is the focus of star schema, like price, discount values, number of units sold, dollar value of sales, and so on. The fact table measure values often represent summarized and aggregated data. Also a fact table is usually very large in terms of rows.

Figure 2 shows a symbolic star schema ERD with four dimension tables and a fact table. Each table has a primary key. The dimension table includes attributes that are the basis for establishing the criteria for retrieving fact measure value. Typically a star schema always includes a time dimension to provide a time perspective during the analysis. Each row of the fact table represents a star schema instance.

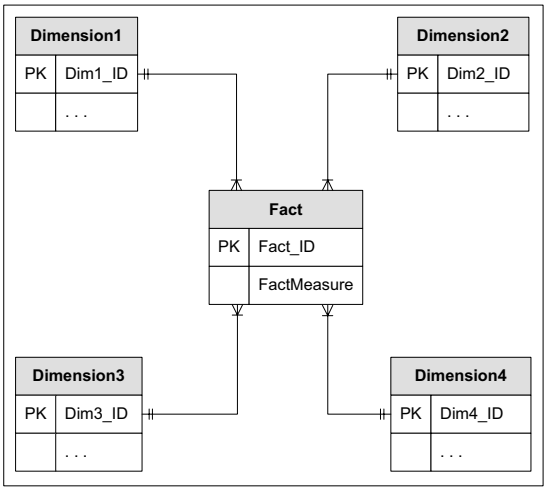

Fig. 2 Symbolic Star Schema ERD

B. Representation of Star Schema Instance as a Business Rule

Merging of business rules with star schema is enabled through the concept of star schema instance. This merging occurs as follows:

1. Each dimension is a constraint in the business rule.
2. Each fact is a constraint in the business rule.
3. Each action refers to the combination of fact and dimensions related to a star schema instance.

A typical business rule for star schema instance can be expressed symbolically as:

```
IF      constraint1  operator  value  AND
        constraint2  operator  value  AND ...
THEN action
```

The IF section represents the attributes of the tables participating in the star schema that need to be validated for the desired action. In the IF section, the *constraint1*, *constraint2*, and so on represent the dimension and fact table attributes, the *operator* is the conditional operator (like =, >, and so on), and the *value* refers to the corresponding dimension/fact attribute value. The *action* entry in the THEN section can be a narrative conclusion that reflects some decision or knowledge based on the star schema instance value set (as specified in the IF section).

For example, a sample business rule structure for the Figure 2 star schema would be as follows:

```
IF      dimension1.attribute operator  value  AND
        dimension2.attribute  operator  value  AND
        dimension3.attribute  operator  value  AND
        fact.factmeasure operator value
THEN action
```

It is not necessary that all the dimension tables participate in a business rule structure. However, every business rule will contain a constraint clause that includes a fact table attribute with its corresponding operator and value. The number of constraints within a business rule are not limited to the number of dimensions within the star schema. It is possible to have constraint clauses corresponding to different attributes of the same dimension table.

A data warehouse can have many star schema (or cubes) pertaining to the needs for analyzing various business activities. Each star schema cube in this case will have its own set of business rules corresponding to its structure. To facilitate modularization, business rules belonging to a star schema cube can be grouped and referred by a cube area name. A cube area business rules in a way pertain to the analysis associated with a specific star schema cube. For example, there could be one cube area of business rules for the sales analysis star schema cube, another cube area of inventory analysis star schema cube, and so on. Cube area concept in a way acts as a container of business rules belonging to a star schema cube.

C. Nature of Business Rules Representation in Database

Even though business rules are defined by the user in declarative language, within the database, these business rules

can be stored as database tables. Structuring of business rules within a database facilitates efficient processing and management of rules, besides providing better association with star schema structure. The storage of business rules in database is based on the concept of cube areas. Business rules pertaining to a cube area are stored in one table named after the cube area, wherein each row represents the individual rule for that area. Each entry of the rule is stored as an attribute in the table. Each cube area table structure will be similar. The generic structure of cube area table as defined through the SQL Create Table syntax is as follows:

```
create table cube_area
(rule_no number(2) constraint cube_area_pk primary key,
action varchar2(150),
const1 varchar2(150),
const1_op varchar2(20),
const1_value varchar2(150),
. . .);
```

where RuleNo is the sequential number for each rule as well as the primary key; action attribute contains the action clause value of the business rule; while the next set of attributes are repeated for each constraint. Const1 attribute is the first constraint name; const1_op attribute is the first constraint conditional operator; and const1_value attribute is the first constraint value in the business rule. Each constraint attribute entry includes the dimension or fact attribute name.

For example, the table structure for the following two business rules is shown in Table 1. Due to many attribute columns, the attributes are shown horizontally, while the rows are shown vertically.

```
Rule 1: IF   item.itemcategory = 'Printer' AND
             time.timequarter = 'Q2' AND
             store.storecity = 'Kansas City' AND
             sum(salesunits) > 8000
     THEN Q2 on target
Rule 2: IF   item.itemcategory = 'Monitor' AND
             time.timequarter = 'Q1' AND
             store.storecity = 'Kansas City' AND
             customer.custzip = 64530
             avg(salesunits) < 600
     THEN Q1 needs improvement
```

TABLE I
SALES ANALYSIS TABLE

Attributes	Row 1	Row 2
RuleNo	1	2
Action	Q2 on target	Q1 needs improvement
Const1	item.itemcategory	item.itemcategory
Const1_op	=	=
Const1_value	Printer	Monitor
Const2	time.timequarter	time.timequarter
Const2_op	=	=
Const2_value	Q2	Q1
Const3	store.storecity	store.storecity
Const3_op	=	=
Const3_value	Kansas City	Kansas City
Const4	sum(salesunits)	customer.custzip
Const4_op	>	=
Const4_value	8000	64530
Const5		avg(salesunits)
Const5_op		<
Const5_value		600

D. Merging Business Rules with Star Schema

Any retrieval of information from the star schema is accomplished through a program unit (database procedure) utilizing dynamic SQL concepts. Essentially, each business rule is processed through a SQL query using dynamic SQL concepts. Once the query is executed, its results are further processed to validate the business rule. To illustrate the business rule query of a star schema cube, a sample star schema cube ERD is shown in Figure 3.

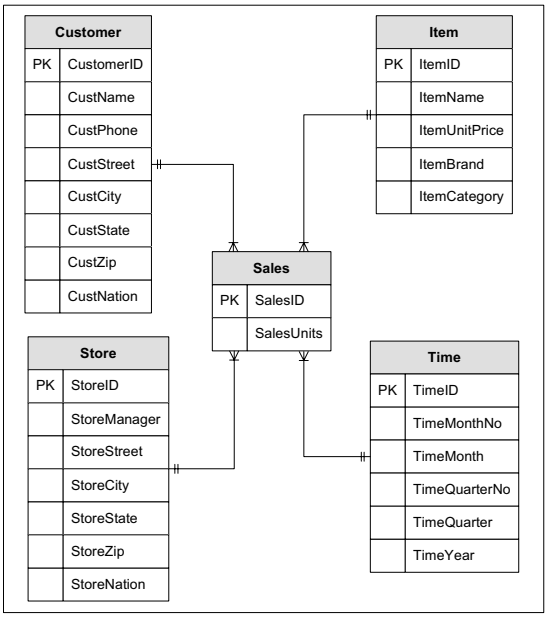

Fig. 3 Star Schema Cube ERD

In Figure 3, Customer, Item, Store, and Time tables represent the dimensions, while Sales is the fact table. Dimension table attributes define the criteria on the basis of which query will be formed. A typical query toward the star schema will be:

For customer attribute, product attribute, time attribute, store attribute find the sales (SalesUnits) value.

where each instance of the Sales fact represents the linking of dimension tables with the fact table.

The matching of a business rule with star schema instance through a SQL query occurs as follows:

1. The rule structure pertaining to the dimension constraints defines the constituents of the WHERE clause in the query.
2. The fact constraints define the constituents of the SELECT clause.
3. The query will always contain a join of all the tables involved in the star schema. The symbolic structure of the query will be as follows:

```
select fact-constraint
from fact-table, dimension-table1, dimension-table2, . . .
where dimension-constraint1 constraint1-op constraint1-value and dimension-constraint2 constraint2-op constraint2-value
and . . . (other dimension constraint expression)
and fact-table.attribute = dimension-table1.attribute
and fact-table.attribute = dimension-table2.attribute
and . . . (other join conditions)
```

For example, the Row 1 business rule in Table 1 can be expressed in SQL query as follows:

```
select sum(salesunits)
from sales, item, time, store, customer
where item.itemcategory = 'Printer'
and time.timequarter = 'Q2'
and store.storecity = 'Kansas City'
and sales.itemid = item.itemid
and sales.timeid = time.timeid
and sales.storeid = store.storeid
and sales.custid = customer.custid
```

where item.itemcategory = 'Printer' and time.timequarter = 'Q2' and store.storecity = 'Kansas City' are dimension constraints, while sales.itemid = item.itemid and sales.timeid = time.timeid and sales.storeid = store.storeid and sales.custid = customer.custid are join conditions.

The matching of the constraint containing the fact value is checked last after the query has successfully executed. Such matching now determines the status of rule validation. In the above example, the sum(salesunits) value is checked next to see whether it is greater than 8000 as listed in the business rule through "sum(salesunits) > 8000" constraint. This final matching eventually determines whether action should be followed or not.

III. WEB PROTOTYPE

A prototype to illustrate the representation of business rule in database, and its utilization to query a star schema is shown through a Web interface using Oracle's PL/SQL server pages technology. The prototype uses the star schema shown in Figure 3. The table rows with sample values for the Figure 3 star schema are shown in Table 2 through Table 6.

TABLE II
ITEM TABLE

Itemid	Itemname	Item Unit Price	Item Brand	Item Category
101	17 inch Color Monitor	369	Dell	Monitor
102	19 inch Color Monitor	319	HP	Monitor
103	R3000 Color Laser Printer	699	Dell	Printer
104	10 Foot Printer Cable	12	D-Link	Accessories
105	8-Outlet Surge Protector	14.99	HP	Accessories
106	CVP Ink Jet Color Printer	99	HP	Printer
107	Color Ink Jet Cartridge	38	Dell	Printer
108	36-Bit Color Scanner	199.99	HP	Accessories
109	Black Ink Jet Cartridge	25.69	Dell	Printer
110	Battery Back-up System	89	D-Link	Accessories

TABLE III
STORE TABLE

Storeid	Store Manager	Storecity	Store Zip	Store Nation
101	Jack Ellis	Kansas City	64530	USA
102	Abena Mackey	Kansas City	64530	USA
103	Tracy Manner	St. Louis	63140	USA
104	Mark Roberts	St. Louis	63140	USA

TABLE IV
CUSTOMER TABLE

Custid	Custname	Custcity	Cust Zip	Cust Nation
101	Lacy George	Kansas City	64530	USA
102	Sania Raina	Kansas City	64530	USA
103	Larry Ellison	St. Louis	63140	USA
104	Robert Phillips	St. Louis	63140	USA

TABLE V
TIME TABLE

Timeid	Time month no	Time month	Time quarter no	Time quarter	Time year
101	1	01-JAN-07	1	Q1	2006
102	2	01-FEB-07	1	Q1	2006
103	3	01-MAR-07	1	Q1	2006
104	4	01-APR-07	2	Q2	2006
105	5	01-MAY-07	2	Q2	2006

106	6	01-JUN-07	2	Q2	2006
107	7	01-JUL-07	3	Q3	2006
108	8	01-AUG-07	3	Q3	2006
109	9	01-SEP-07	3	Q3	2006
110	10	01-OCT-07	4	Q4	2006
111	11	01-NOV-07	4	Q4	2006
112	12	01-DEC-07	4	Q4	2006

TABLE VI
SALES TABLE

Salesid	Itemid	Storeid	Custid	Timeid	Salesunits
100	101	101	101	101	500
101	102	102	102	102	550
102	103	103	103	103	1500
103	104	104	101	104	3500
104	105	101	102	105	8500
105	106	102	103	106	4500
106	107	103	101	107	6500
107	108	104	102	108	5500
108	109	101	103	109	6000
109	110	102	101	110	300
110	101	103	102	111	200
111	102	104	103	112	8200
112	101	101	101	101	500
113	102	102	102	102	550
114	103	103	103	103	1500
115	104	104	101	104	3500
116	105	101	102	105	8500
117	106	102	103	106	4500
118	107	103	101	107	6500
119	108	104	102	108	5500
120	109	101	103	109	6000
121	110	102	101	110	300
122	101	103	102	111	200
123	102	104	103	112	8200

To simplify the conceptual implementation, Oracle standard SQL is utilized instead of Oracle data warehouse specific statements like CREATE DIMENSION, and so on. The Oracle data warehouse statements are more attuned for efficiency during warehouse setup and working.

There are two business rules for the star schema as outlined in Table 1(sales_analysis table). The SQL to store the business rules in database is shown below.

```
create table sales_analysis
(ruleno integer constraint sa_pk primary key,
action varchar2(150),
const1 varchar2(150),
const1_op varchar2(20),
const1_value varchar2(150),
const2 varchar2(150),
const2_op varchar2(20),
const2_value varchar2(150),
const3 varchar2(150),
const3_op varchar2(20),
const3_value varchar2(150),
const4 varchar2(150),
const4_op varchar2(20),
const4_value varchar2(150),
const5 varchar2(150),
const5_op varchar2(20),
const5_value varchar2(150));
```

The prototype consists of two Web pages, and is limited to one cube area. The interaction of the two Web pages within the Web architecture is shown in Figure 4. PL/SQL server pages are stored as Web procedures within the Oracle database.

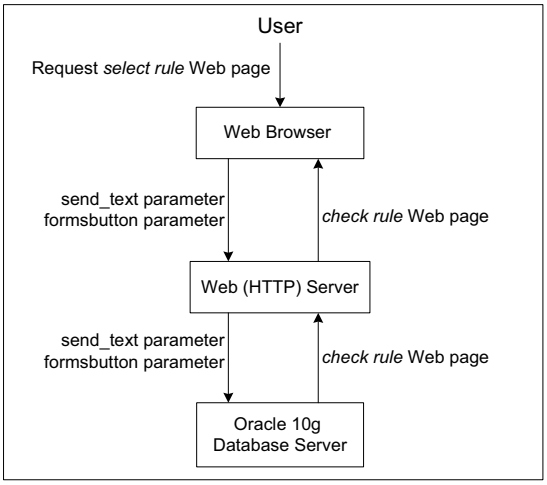

Fig. 4 Web Architecture for PL/SQL Server Pages

The user requests for the first Web page titled select rule. This page displays the existing business rules for selection. Figure 5 shows a view of the select rule Web page. The select rule Web page is generated through a Web procedure also titled “select_rule” as listed in Appendix-A.

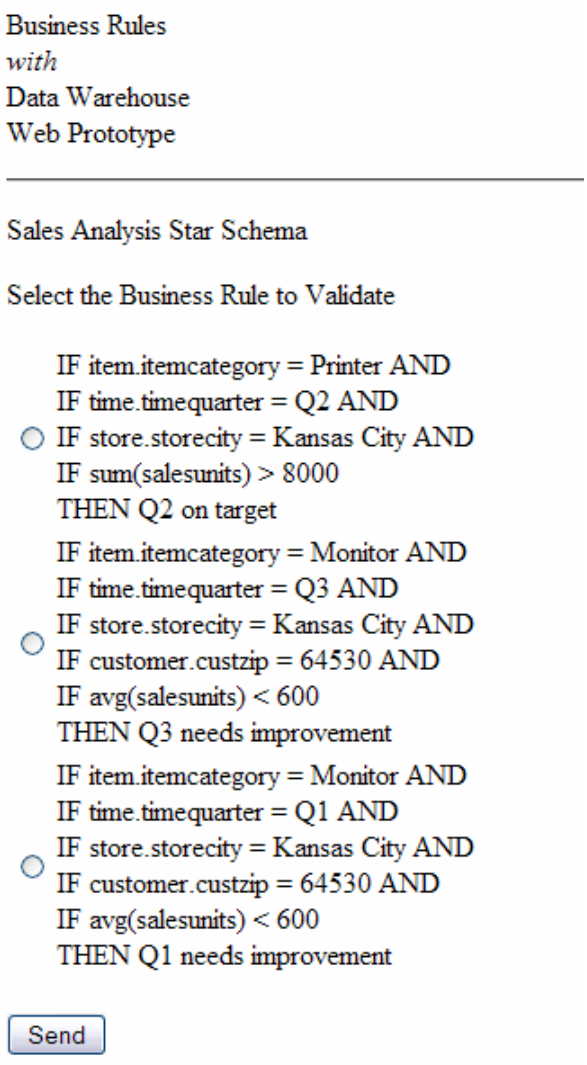

Fig. 5 Select Rule Web page in Browser

The response page titled check rule validates the selected rule displayed on the select rule page. Figure 6 shows the response page in browser when the rule is valid, while Figure 7 shows the response page in browser when the rule is invalid.

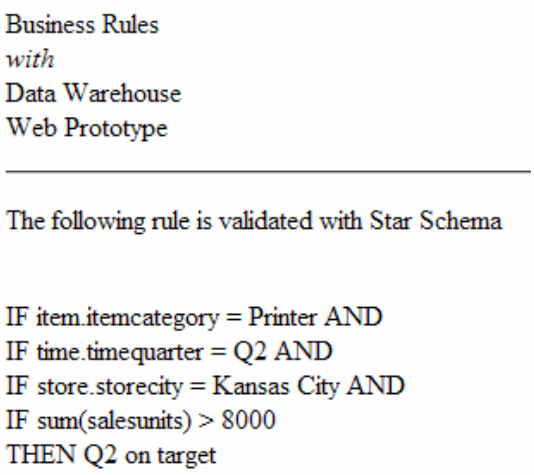

Fig. 6 Check Rule Web page in Browser showing Rule Validation

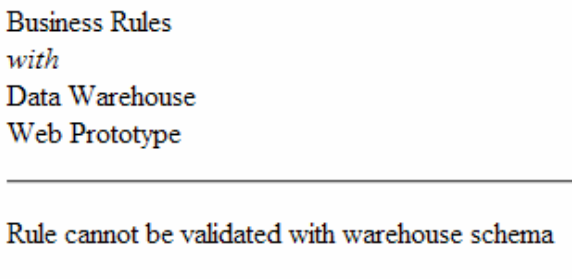

Fig. 7 Check Rule Web page in Browser showing Invalid Rule

The check rule Web page is generated through a Web procedure also titled “check_rule” as listed in Appendix-B. The check_rule procedure logic of business rule validation with the star schema is shown in Figure 8.

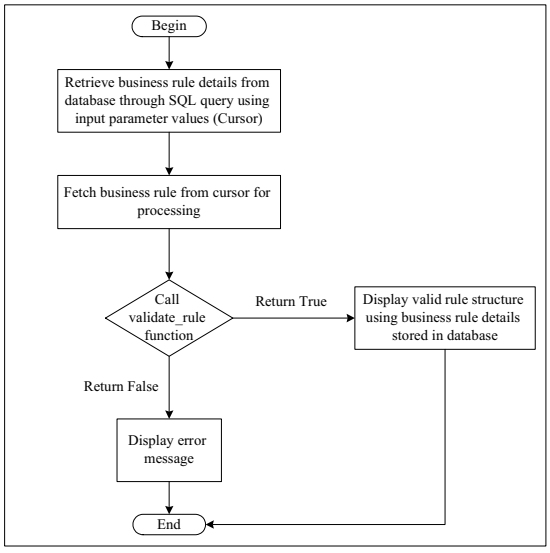

Fig. 8 check_rule procedure logic for business rule validation

The check_rule Web procedure utilizes a database function validate_rule to complete the processing. The logic of the function validate_rule is shown in Figure 9, while its details are listed in Appendix-C.

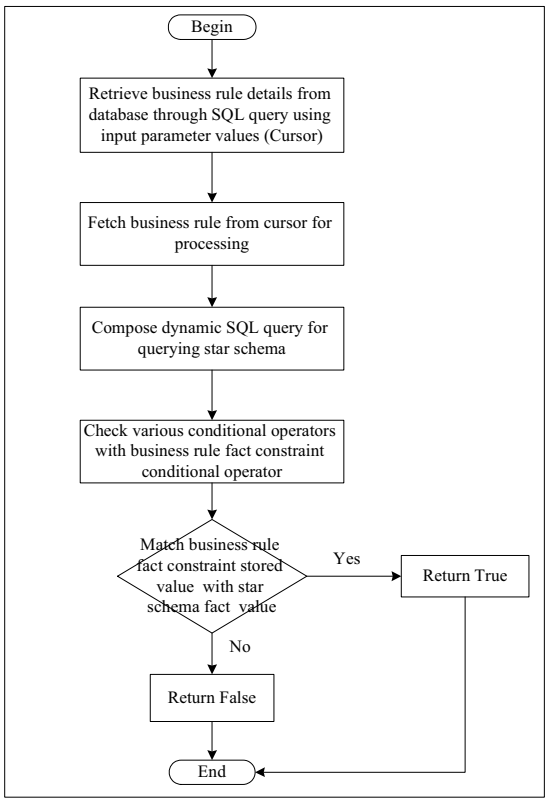

Fig. 9 validate_rule function logic

IV. CONCLUSION

Use Data warehouse star schema is a very powerful concept to perform business analysis. Typically such analysis is performed through highly specialized applications which include drill-down and roll-up features. Business rules represent business knowledge necessary for organization working. Use of business rule concepts to analyze data warehouse extends the scope of business rules beyond transactional applications.

This paper is an attempt to illustrate how business rules can be applied to data warehouse star schema setup. Further research is in progress to extend the prototype features. This involves (i) providing the ability to validate business rules on other data warehouse schemas like snowflake and constellation, (ii) incorporate additional complexity in rule specification, (iii) explore the utility of rule chain levels, and (iv) including additional data warehouse activity.

APPENDIX-A

```
<%@page language="PL/SQL"%>
<%@plsql procedure="select_rule"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
<head>
<title>Select Rule</title>
</head>
<body>
<div><p>Business Rules <br>
<i>with</i><br>
Data Warehouse<br>
Web Prototype</p>
<hr /></div>
<!-- Start Page Content -->
<p>Sales Analysis Star Schema</p>
<p>Select the Business Rule to Validate</p>

<form action="check_rule" method="get">

<!-- Set of two radio buttons -->
<table>
<%for rule_row in (select * from sales_analysis)
loop%>
<tr><td><input type="radio" name="send_text"
value="<%=rule_row.ruleno%>" /></td>
<td><%='IF '||rule_row.const1||' '||rule_row.const1_op||'
'||rule_row.const1_value||' AND'%><br />
<%='IF '||rule_row.const2||' '||rule_row.const2_op||'
'||rule_row.const2_value||' AND'%><br />
<%='IF '||rule_row.const3||' '||rule_row.const3_op||'
'||rule_row.const3_value||' AND'%><br />
<%='IF '||rule_row.const4||' '||rule_row.const4_op||'
'||rule_row.const4_value%>
<%if rule_row.const5 is not null then%>
<%=' AND'%><br />
<%='IF '||rule_row.const5||' '||rule_row.const5_op||'
'||rule_row.const5_value%><br />
<%end if;
if rule_row.const5 is null then%>
<%=' '%><br />
<%='THEN '||rule_row.action%><br />
<%else%>
<%='THEN '||rule_row.action%><br />
<%end if;%></td></tr>
<%end loop;%>
</table>
<br />

<!-- Set of Submit and Reset buttons -->
<input type="submit" name="FormsButton1"
value="Send"/>
</form>
<!-- End Page Content -->
```

```
</body>
</html>
```

APPENDIX-B

```
<%@page language="PL/SQL"%>
<%@plsql procedure="check_rule"%>

<!-- Input parameters received from the first Web page for
selected rule -->

<%@plsql parameter="send_text" default="null"%>
<%@plsql parameter="formsbutton1" default="null"%>

<!-- retrieve rule details from the database -->
<%!cursor rule_cur is
select * from sales_analysis
where ruleno = send_text;
rule_row rule_cur%rowtype;%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">

<head>
<title>Check Rule</title>
</head>
<body>
<div><p>Business Rules <br>
<i>with</i><br>
Data Warehouse<br>
Web Prototype</p>
<hr /></div>
<!-- Start Page Content -->
<% open rule_cur;
fetch rule_cur into rule_row;

<!-- Call function validate_rule for validation -->

if validate_rule(send_text) then%>
<p>The following rule is validated with Star
Schema</p><br />

<!-- Display valid rule in the Web page -->

<%='IF '||rule_row.const1||' '||rule_row.const1_op||' '||
rule_row.const1_value ||' AND'%><br />
<%='IF '||rule_row.const2||' '||rule_row.const2_op||' '||
rule_row.const2_value||' AND'%><br />
<%='IF '||rule_row.const3||' '||rule_row.const3_op||' '||
rule_row.const3_value||' AND'%><br />
 <%='IF '||rule_row.const4||' '||rule_row.const4_op||' '||
rule_row.const4_value%>

<%if rule_row.const5 is not null then%>
    <%=' AND'%><br />
    <%='IF '||rule_row.const5||' '||rule_row.const5_op||' '||
rule_row.const5_value%><br />
<%end if;

if rule_row.const5 is null then%>
    <%=' '%><br />
    <%='THEN '||rule_row.action%><br />
    <%else%>
    <%='THEN '||rule_row.action%><br />
<%end if;%>
<%else%>

<!-- Display error message -->

  <p>Rule cannot be validated with warehouse
schema</p>
<%end if;%>
<%close rule_cur;%>
</body>
</html> acknowledgment.
```

APPENDIX-C

```
Appendixes, create or replace function validate_rule
(send_text varchar2)
return boolean as

/* retrieve rule details from the database */
cursor rule_cur is
select * from sales_analysis
where ruleno = send_text;
rule_row rule_cur%rowtype;
str varchar2(2000);
comp_value integer;
fact_op varchar2(15);
fact_value integer;

begin
open rule_cur;
fetch rule_cur into rule_row;

/* compose dynamic SQL query for querying the star
schema */
if rule_row.const5 is null then

str := 'select '||rule_row.const4||' from sales, item, store,
customer, time where sales.itemid = item.itemid and
sales.storeid = store.storeid and sales.timeid = time.timeid and
sales.custid = customer.custid
and  '||rule_row.const1||' = :i and '||rule_row.const2||' = :j and
'||rule_row.const3||' = :k';

execute immediate str into comp_value using
rule_row.const1_value, rule_row.const2_value,
rule_row.const3_value;

fact_op := rule_row.const4_op;
fact_value := rule_row.const4_value;
```

```
   else

   str := 'select '||rule_row.const5||' from sales, item, store,
customer, time where sales.itemid = item.itemid and
sales.storeid = store.storeid and sales.timeid = time.timeid and
sales.custid = customer.custid
   and  '||rule_row.const1||' = :i and '||rule_row.const2||' = :j and
'||rule_row.const3||' = :k and '||rule_row.const4||' = :l';

   execute immediate str into comp_value using
rule_row.const1_value, rule_row.const2_value,
rule_row.const3_value, rule_row.const4_value;
   fact_op := rule_row.const5_op;
   fact_value := rule_row.const5_value;
   end if;

   /* Check the fact condition in the business rule with the
value returned from the star schema to validate the business
rule */

   if (fact_op = 'is') or (fact_op = '=') then
      if comp_value = fact_value then
         return true;
      else
         return false;
      end if;
   end if;
   if (fact_op = 'is greater than') or (fact_op = '>') then
      if comp_value > fact_value then
         return true;
      else
         return false;
      end if;
   end if;
   if (fact_op = 'is less than') or (fact_op = '<') then
      if comp_value < fact_value then
         return true;
      else
         return false;
      end if;
   end if;
   if fact_op = 'is between' then
      if (fact_op >= fact_value) and (fact_op <= fact_value)
then
         return true;
      else
         return false;
      end if;
   end if;
   end;
```

REFERENCES

[1] Y. Amghar, M. Meziane, and A. Flory, "Modeling of Business Rules For Active Database Application Specification," in *Advanced Topics in Database Research*, K. Siau, Ed., Hershey, PA: Idea Group Publishing, 2002, pp. 135-156.
[2] S. Boardman, M. Caffrey, S. Morse, and B. Rosenzweig, *Oracle Web Application Programming for PL/SQL Developers*, Upper Saddle River, NJ: Prentice-Hall, 2003.
[3] C.J. Date, *What Not How: The Business Rules Approach to Application Development*, Reading, MA: Addison-Wesley, 2000.
[4] C. J. Date and H. Darwen,, *Foundation for Future Database Systems: The Third Manifesto*, 2nd ed., Reading, MA: Addison-Wesley, 2000.
[5] J. Dyche, *e-Data*, Reading, MA: Addison-Wesley, 2000.
[6] B.V. Halle, *Business Rules Applied*, New York, NY: John Wiley & Sons, 2002.
[7] D.C. Hay, "A Repository Model - Business Rules - Part I (Structural assertions and derivations)," *The Data Administration Newsletter*, Issue 19, January 2002. Available: http://www.tdan.com
[8] D.C. Hay, "A Repository Model - Business Rules - Part II (Action Assertions)," *The Data Administration Newsletter*, Issue 20, April 2002. Available: http://www.tdan.com
[9] D.C. Hay, "Modeling Busines Rules: What Data Models Do," *The Data Administration Newsletter*, Issue 27, January 2004. Available: http://www.tdan.com
[10] R. Kaula, *Oracle 10g: Developing Web Applications with PL/SQL Server Pages*, New York, NY: Mc-Graw-Hill, 2006.
[11] R. Kimball, *The Data Warehouse Toolkit*, New York, NY: John Wiley & Sons, 2002.
[12] S. Ram and V. Khatri, "A comprehensive framework for modeling set-based business rules during conceptual database design," *Information Systems*, Vol. 30, pp. 89-118, 2005.
[13] R.G. Ross, *The Business Rule Book: Classifying, Defining, and Modeling Rules*, 2nd ed., Boston, MA: Database Research Group, 1997.

Dr. Rajeev Kaula is presently a Professor in Computer Information Systems at Missouri State University, Springfield, Missouri (USA). He completed his MBA from Banaras Hindu University (India), MS from Roosevelt University (USA) and Ph.D from State University of New York, Binghamton (USA). He has about 4 years of industry experience, and about 20 years of experience in teaching and research. His fields of interest include database applications, business rules working, data warehouse, business intelligence, business process management, computer integrated manufacturing, and open information systems. He has also written books on Oracle PL/SQL Server Pages and Open Information Systems. He has more than 40 publications to his credit in international journals and conferences.