DISCOVERY OF PROBABLE DIMENSIONS & FACTS FROM RAW DATA AVAILABLE IN THE FILE SYSTEMS AND ORACLE ® RELATIONAL DATABASE SCHEMAS FOR A MULTI-DIMENSIONAL DATA-WAREHOUSING APPLICATION

Goparaju V. Ramesh[1], Mogalla Shashi[2] and KVSS Rajeswara Rao[3]

[1&3]IT Department, Visakhapatnam Steel Plant, Visakhapatnam-530 031, India
[2]Deptt. of CS & SE, College of Engineering, Andhra University, Visakhapatnam, India

ABSTRACT

Multidimensional Data-warehousing applications play prominent role in facilitating business intelligence, and subsequent strategic decision making for the industry. It provides a good starting point for data-mining applications, which cater to different segments of industry, academia, research and government sectors. This paper provides for discovery of probable dimensions and facts from typical data-stores of any organisation, be they file-systems or Oracle ® relational database schemas. In the case of file-systems, a preprocessing step is executed to port the data in the file-system into the Oracle ® relational database. The dimensions are then discovered based on the attribute data-types; relational constraints, wherever existing; based on domain-consistency, whereas, the facts are discovered based on the discovered dimensions and attribute data-types.

KEYWORDS: *Data-Warehousing Application, Oracle ® Relational Database, Discovery of Dimensions, Discovery of Facts, File System*.

I. INTRODUCTION

Nowadays, it is the relational databases that form the major source of data in any business application system. The usage of files for storing data has past its prime some time ago. Most of the organizations which were using File-systems in their legacy mainframes, mini-computers have migrated their data into relational databases. Many vendors have provided software tools to port these data into relational databases. In such a scenario, the data existing in relational databases assume paramount importance in any organization in all its business spheres.

This paper provides for auto-discovery and extraction of likely dimensions and facts for multi-dimensional data-warehousing applications, which may be of interest to the business analysts, strategic thinkers, etc. in any organization. Generally the dimensions and facts are framed by business experts based on their experience, their understanding of IT application systems, business rules, etc. Sometimes, it becomes difficult to get quality experts, who can do this job. Sometimes, the IT application systems become so big and heterogeneous that it becomes very difficult even for experts to find out where to start. This is a major problem in those organizations, where systems grow in a haphazard manner at different points of time. The solution provided in this paper extracts the potential dimensions and facts from Oracle ® databases as well as file systems. Otherwise, it provides a starting point for even casual expert to make his way clear for identifying the underlying dimensions and facts. Wherever file system forms part of the corporate data-bank, a preliminary process has to be carried to port the entire file-data into relational database.

The manuscript is ordered as follows: The concept & importance of dimensions and facts for a data-warehouse are discussed in the section of "Dimensions & Facts in Data-warehousing". The main aim of this paper is to discover out the dimensions & facts from the Relational Databases. Therefore, unless the characteristics are delineated in the beginning itself, this paper does not do justice. The section characteristics of Dimensions in terms of their representation in Databases" does the same. The procedures adopted for the discovery of dimensions and facts have been addressed in detail in sections The Procedure adopted for discovery of dimensions & The Procedure adopted for discovery of Facts / Measures. The results obtained from the system, developed and deployed using the design discussed in previous sections, are tabulated and discussed in the section Tabulation of results. The paper has been concluded in Conclusions section and finally the references have been shown in section References.

II. DIMENSIONS & FACTS IN DATA-WAREHOUSING

Dimensions and facts form part of what a data-warehouse contains. The dimensions provide the ways in which a fact/measure is analyzed. It generally indicates the master data, like supplier, time, region, etc. There are different hierarchies built in the dimensions. The fact table contains the actual numerical measures that need to be analyzed across different dimensions, like the sales value, purchase volume, cash disbursed, etc.
The authors are practicing and academic specialists in building data-warehouses from operational databases. The authors have observed following characteristics and properties of different dimensions and facts while building efficient, effective and large data-warehouses from operational databases. The typical facts & measures identified by the authors during their regular as well as research work in the industry/organisations are as follows:

Table 1. Typical Facts & Measures.

Business Domain	Facts / Measures to explore	Dimensions
Maintenance, Repair & Operation	Operational Efficiency of Equipment, Maintenance History of Equipment, Productivity of different production departments, Product Quality parameters, Production parameters, etc	Product, Manufacturing section, Equipment type, time, etc.
Marketing & Sales	Fast moving products, Well run branches/regions, Valued customers, Net Sales Realization Market trends, etc.	Product, Region, Customer, Time, etc.
Materials	Vendor Classification, Fast moving spares, Economic order quantities, Estimates for future procurements, etc. Lead-time	Supplier, Region, Material, Time, etc.
Human Resources	Employee Commitment, Per capita pay & perks, Employee Medical history, etc.	Employee Category, Employee Age, Employee Health Category, Time, etc.
Financials	Different Financial indices, Costing Sensitivity analysis, Cash Flow, Fund flow variances, etc.	Section, Account type, Schedule, Time, etc.

III. THE CHARACTERISTICS OF DIMENSIONS IN TERMS OF THEIR REPRESENTATION IN DATABASES

Dimensions are generally alpha-numeric in nature. Even if they contain purely numerals, they are declared as alphanumeric. E.g., the value '000006' is, in case, stored using numeric datatype attribute, then it is stored as '6', whereas stored as '000006' in alphanumeric fields.
Dimensional fields are generally found in the referential integrity fields in the relational database systems that are perfectly maintained.

Dimensional field identifiers are generally not long. An organizational constant can be arrived based on empirical evidence for the maximum length.
The dimensional field identifiers generally exist in more than one table in the relational databases or in file system. For example as shown in Figure-1, the Supplier_id is found in Supplier Master table as well as Purchase Order transaction table as shown below. Whereas the non-identifier fields, like Supplier name figures only in the Supplier Master table. For that matter, even invoice quantity (a field which may be of interest for fact/measure, which is discussed later) is found nowhere except in its transaction table, generally named as Sales Transactions table.

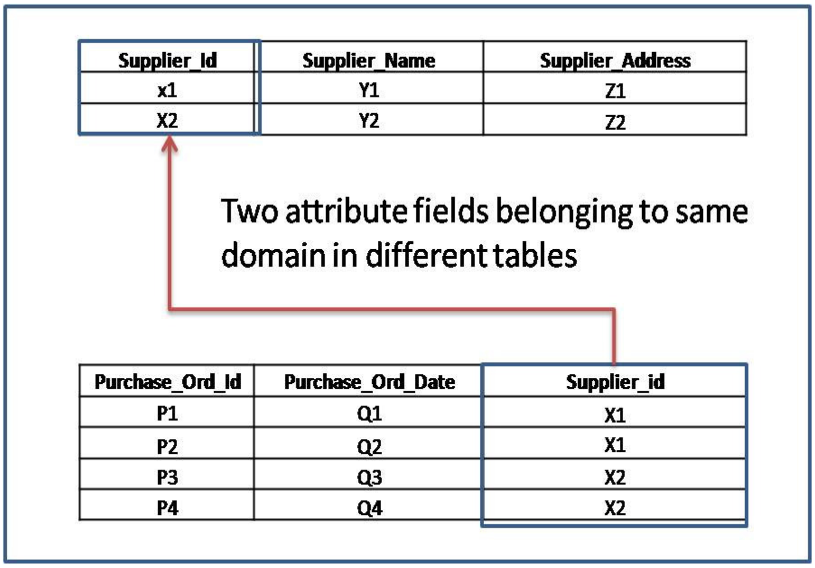

Figure 1. Referential Integrity pointing to a different table in relational databases

The domain consistency exists for dimensional identifier fields. For example, if supplier_code ranges from '000001' to '999999', there will not be any table in the entire system, which has supplier code as its field, and whose domain spans beyond these two lower and upper bounds.

There is a chance of existence of field with referential integrity, pointing to same table as shown in Figure-2. For example, the Boss_employee_id in a table of employees may refer to the employee_id of the same table with "On delete set null" referential constraint as shown below. In the following diagram, X1 is the CEO and there is no one above him in the organizational hierarchy. Therefore, his boss_employee_id is null. Whereas, for X2, X1 is boss. Since, X1 is at first an employee of the organization, a record with him as the primary key should exist in the same table.

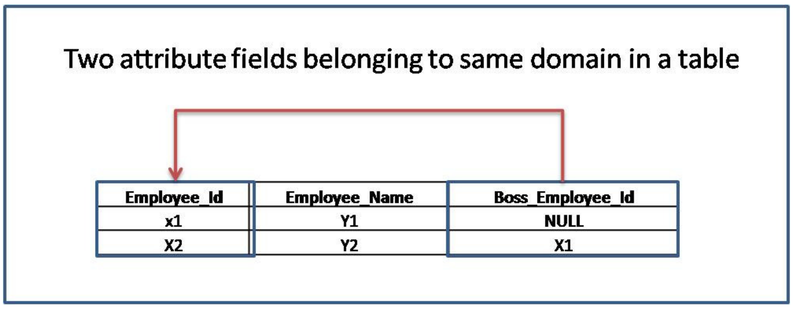

Figure 2. Referential Integrity pointing to a field in the same table in relational databases

This feature may exist without referential integrity also in not-so-perfectly-maintained database system. In such cases, the presence of domain consistency across two attributes of same table is employed to find the real dimensional-identifier, which is discussed elaborately in the subsequent section.
The database table that houses maximum tuples, having distinct identifier can be considered the master table for dimensional identifier.

IV. The Procedure Adopted for Discovery of Dimensions

Based on the premises delineated in the earlier section, the following procedure has been created for discovering dimensions from an organizational database consisting of Oracle ® relational database , herein after referred to as Relational Database System and/or File System
This procedure consists of following steps as shown in the Figure-3 below:

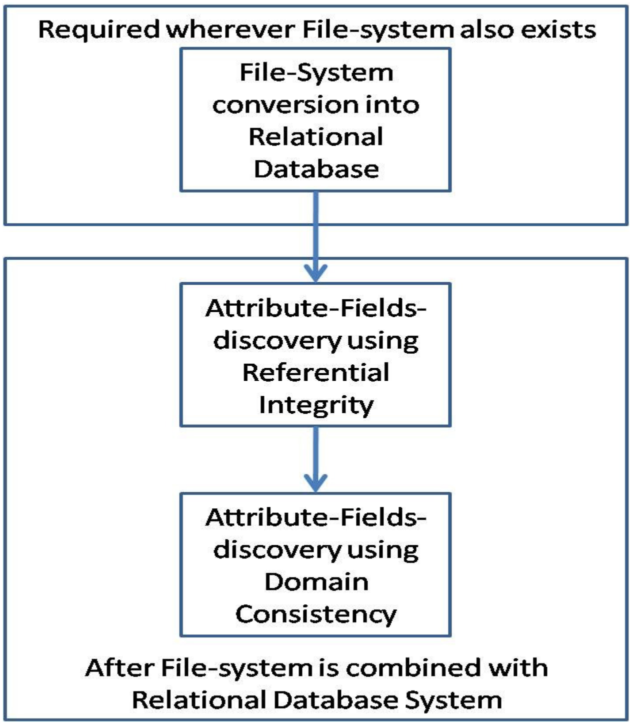

Figure 3. Schematic diagram of procedure for the discovery of dimensions from Relational Databases

4.1 File-System Conversion into Relational Database

This step is needed for only those organizations, which have file-system either alone or in combination with relational database system. In this step, the entire file-system is ported into relational databases. This can be effected using many techniques. Many vendors provide simple tools to accomplish this task. Or, any custom-made application can be coded, which takes, as input, the row length, delimiters of file system, and create tables in the relational database systems using dynamic sql techniques. The datatypes of the fields also can be customized using different functions like is numeric, is logical, etc. and perfectly build a relational database system out of file-system.
The algorithm (suiting to Oracle PL/SQL):

```
Get all file names into a table TOFN
Open cursor CUR_TOFN for selecting file-names
While not end of cursor CUR_TOFN
        Open file in read mode
        Find out the column data types
        Create Oracle table for each file based on column data types
End while
Close cursor CUR_TOFN

Open cursor CUR_TOFN for selecting file-names
While not end of cursor CUR_TOFN
        Open file in read mode
        While not end-of-file
                Insert record column wise into the corresponding Oracle table
        End while
```

End while
Close cursor CUR_TOFN

4.2 Attribute-Fields-discovery using referential Integrity

This step is very useful wherever the database systems are perfectly maintained using referential key integrity. The attribute-fields belonging to dimensions can be found out based on the referential integrity constraint using the catalogue tables/views provided by Oracle ®, like USER_CONSTRAINTS & USER_CONS_COLUMNS, the views provided by Oracle System. This step takes care of perfectly maintained relational databases.

The algorithm (suiting to Oracle PL/SQL):

```
Define cursor CUR_CONS with oracle catalogue view USER_CONSTRAINTS,
USER_CONS_COLUMNS having constraint type as "Primary"
Define cursor CUR_COLS with oracle catalogue view USER_CONS_COLUMNS,
USER_TAB_COLS
OPEN CUR_CONS
OPEN CUR_COLS
While not end of cursor CUR_CONS
        Get table_name, constraint_name
        While not end of cursor CUR_COLS
                Get column_name, column_data_type
                Insert into dimensions table with column_name, data_type
        End while
End while
```

4.3 Attribute-Fields-Discovery using domain consistency

This step takes care of any type of database system. The following diagram at Figure 4 depicts the process followed in a schematic manner

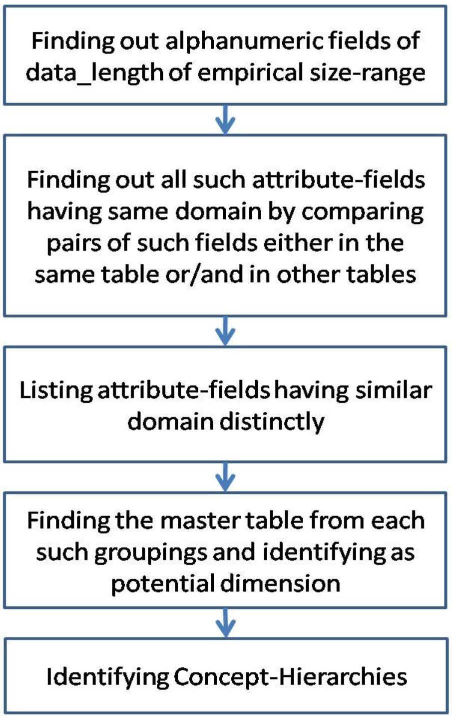

Figure 4. Schematic diagram showing the discovery of Attribute fields using domain consistency

4.3.1 Finding out alphanumeric fields of data_length of empirical size-range :

The attribute-fields of alphanumeric fields of given size-range can be found out using catalog view, USER_TABLES & USER_TAB_COLS in Oracle

Finding out all such attribute-fields having same domain by comparing pairs of such fields either in the same table or/and in other tables
The fileds so identified in the previous step are paired with every other such field and the domain of the fields is found out using dynamic sql query techniques. A small glimpse is given here at Figure 5 to give set-theory-representation of domain consistency.

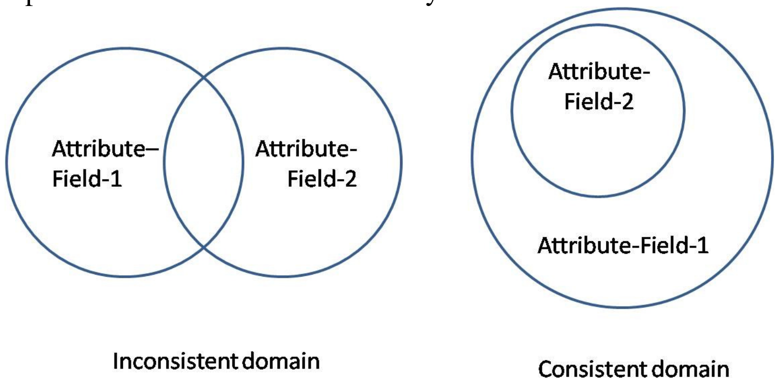

Figure 5. Schematic diagram showing the inconsistent domains and consistent domains

4.3.2 Listing out domain-consistent attribute-fields

After the previous step is completed, all the fields which are similar in domain are grouped separately as shown in the following diagram at Figure 6.

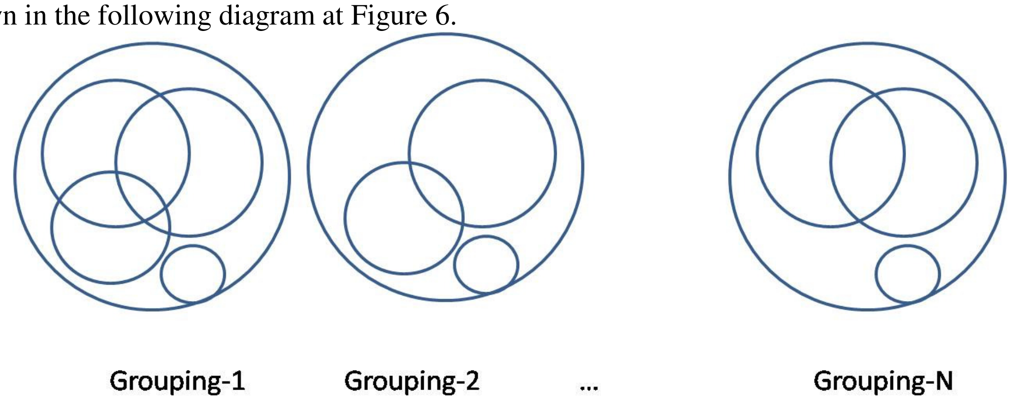

Figure 6. Schematic diagram showing the groupings of consistent domains.

4.3.3 Finding out the master table from each such group

The table that contains maximum number of such distinct identifiers is generally considered the dimensional master. Such table also may contain the description of the dimension.
Identifying concept hierarchies:
The concept hierarchy within a dimension can be found out in the following ways

- Find out the distance between minimum and maximum values of the field. If the distance is much much higher than the number of distinct attributes, it indicates there is some intelligence built in the code, which signifies the concept hierarchy. Eg., 200001,300002 are the two tuples of an attribute, then their difference is 100001, where as there are only two distinct numbers in the database. It signifies that the first digit has some codification. The first digit ('2','3') may refer to a year or a department. It can be further probed by experts.
- Identifying from other attributes. Find out the average of count distinct dimension-identifier field with respect to all other attributes in the table individually. If the average is found to be higher than any empirical constant, then that field is considered to be of higher level to this dimensional field.

The algorithm (suiting to Oracle PL/SQL):

/* The following for checking domain consistency */
Define cursor CUR_NON_NUMERIC_ATTR for selecting table_name, column_name and data_type from USER_TAB_COLUMNS having column_datatype of "char" and "varchar2"

```
Define cursor CUR_NON_NUMERIC_ATTR_COMPARE for selecting  table_name,
column_name and data_type such that all "char, varchar2" data type columns of other than
the table selected using cursor CUR_NON_NUMERIC_ATTR
OPEN CUR_NON_NUMERIC_ATTR
While not end of cursor CUR_NON_NUMERIC_ATTR
        Select  table_name TN1, column_name CN1
        OPEN CUR_NON_NUMERIC_ATTR_COMPARE
        While not end of cursor CUR_NON_NUMERIC_ATTR_COMPARE
                Select  table_name TN2, column_name CN2
                Check the intersection of both the domains
                If both CL1, CL2 belong to same domain
                        Insert (TN1, CL1), (TN2,CL2) into output_table with same
                        grouping_no
                End if
        End while
        End cursor CUR_NON_NUMERIC_ATTR_COMPARE
End while
End cursor CUR_NON_NUMERIC_ATTR
```

V. THE PROCEDURE ADOPTED FOR DISCOVERY OF FACTS / MEASURES

The important characteristics as observed by the authors for facts/measures are mentioned hereunder:

- The datatypes of facts/measures are always numeric in nature
- Some of the facts are to be derived from identifiers of date datatypes. Example for the latter is "Lead-time in a purchase transaction" cannot be directly found in any Materials Management Applications. It is derived as difference between two dates, viz., Purchase Order Date and Material Receipt Date.
- The facts derived from date-related identifiers do not necessarily have those related dates in the same table. But, there will be some common other identifiers in their respective tables.
- The most important commonality found in such facts is that the date found in one table always precedes the date found in the other table for the related tuples. For example, delivery date cannot be prior to the Purchase order date. The material inspection date cannot be earlier to the material receipt date.

The procedure adopted for period-calculation is as follows:

- Find date identifiers in same table
- Eliminate audit-trail fields from the table
- Find those tables, where two or more date fields exist
- Find out for all the tuples, one date field is always more than the other or vice-versa
- In case true, then the difference can be considered as one fact/mesure

The algorithm (suiting to Oracle PL/SQL):

```
/* The following algorithm is for numeric facts directly from tables */

Define cursor CUR_NUMERIC_ATTR for selecting  table_name, column_name and data_type
from USER_TAB_COLUMNS having column_datatype of "number"

OPEN CUR_NUMERIC_ATTR
While not end of cursor CUR_NUMERIC_ATTR
    Select  table_name TN1, column_name CN1
            Insert (TN1, CL1) into output_table
    end while
close cursor CUR_NUMERIC_ATTR
```

```
/* The following algorithm is for date related fields from same table */

Define cursor CUR_DATE_ATTR for selecting  table_name, column_name from
USER_TAB_COLUMNS having column_datatype of "date" of same table name

OPEN CUR_DATE_ATTR
While not end of cursor CUR_NUMERIC_ATTR
    Select  table_name TN1, column_name CN1
            Insert (TN1, CL1) into intermediate_table with same grouping_no
    end while
close cursor CUR_DATE_ATTR

Find actual tuple values of these columns from actual tables of the schema where one column
value is always less than or more than of the other column values
Insert such pair of fields into output_table
```

VI. TABULATION OF RESULTS

Inputs:
All the tables and views of a schema become the input data for this algorithm
In-Process-data-tables
The domain comparison control table:
All the attribute fields that are compared are inserted and verified during the processing of the algorithm. The results are shown in Table 2.

Table 2. Result set of In-Process-data-tables

	TABLE_NAME1	COLUMN_NAME1	TABLE_NAME2	COLUMN_NAME2
1	TMOB_OUT_BOX_BULK	PHONE_NO	VEMP_PHONE_MAS	EMPNO
2	VEMP_PHONE_MAS	EMPNO	TMOB_OUT_BOX_BULK	PHONE_NO
3	TMOB_OUT_BOX_BULK	PHONE_NO	VEMP_PHONE_MAS	TPNO
4	VEMP_PHONE_MAS	TPNO	TMOB_OUT_BOX_BULK	PHONE_NO
5	TMOB_OUT_BOX_BULK	PHONE_NO	VMOB_PHONE_MAS	EMPNO
6	VMOB_PHONE_MAS	EMPNO	TMOB_OUT_BOX_BULK	PHONE_NO
7	TMOB_OUT_BOX_BULK	PHONE_NO	VMOB_PHONE_MAS	TPNO
8	VMOB_PHONE_MAS	TPNO	TMOB_OUT_BOX_BULK	PHONE_NO
9	TMOB_PHONE_MAS	EMPNO	TMOB_STATIC_OUT_BOX	DEPT_CODE
10	TMOB_STATIC_OUT_BOX	DEPT_CODE	TMOB_PHONE_MAS	EMPNO
11	TMOB_PHONE_MAS	EMPNO	TMOB_STATIC_OUT_BOX	EMP_NO
12	TMOB_STATIC_OUT_BOX	EMP_NO	TMOB_PHONE_MAS	EMPNO
13	TMOB_PHONE_MAS	EMPNO	TMOB_STATIC_OUT_BOX	PHONE_NO
14	TMOB_STATIC_OUT_BOX	PHONE_NO	TMOB_PHONE_MAS	EMPNO
15	TMOB_PHONE_MAS	EMPNO	VCOFFEE_PERSONS	EMPNO
16	VCOFFEE_PERSONS	EMPNO	TMOB_PHONE_MAS	EMPNO
17	TMOB_PHONE_MAS	EMPNO	VCOFFEE_PERSONS	ENAME
18	VCOFFEE_PERSONS	ENAME	TMOB_PHONE_MAS	EMPNO
19	TMOB_PHONE_MAS	EMPNO	VEMP_NAME_PHONE_MAS	DEPT_CD
20	VEMP_NAME_PHONE_MAS	DEPT_CD	TMOB_PHONE_MAS	EMPNO
21	TMOB_PHONE_MAS	EMPNO	VEMP_NAME_PHONE_MAS	DEPT_DESC
22	VEMP_NAME_PHONE_MAS	DEPT_DESC	TMOB_PHONE_MAS	EMPNO

23	TMOB_PHONE_MAS	EMPNO	VEMP_NAME_PHONE_MAS	DESIGN_DESC
24	VEMP_NAME_PHONE_MAS	DESIGN_DESC	TMOB_PHONE_MAS	EMPNO
25	TMOB_PHONE_MAS	EMPNO	VEMP_NAME_PHONE_MAS	EMPNO
26	VEMP_NAME_PHONE_MAS	EMPNO	TMOB_PHONE_MAS	EMPNO
27	TMOB_PHONE_MAS	EMPNO	VEMP_NAME_PHONE_MAS	ENAME
28	VEMP_NAME_PHONE_MAS	ENAME	TMOB_PHONE_MAS	EMPNO
29	TMOB_PHONE_MAS	EMPNO	VEMP_NAME_PHONE_MAS	ORG

Notes on the In-process – Domain comparison table:

1. The purpose of this table is to ensure that a pair once checked, should not be repeated again
2. It is a Cartesian product table cutting across pairs of field-attributes of the entire schema.
3. As can be seen from this table-2, the column TABLE_NAME1, COLUMN_NAME1 and TABLE_NAME2, COLUMN_NAME2 refer to two columns of the schema belonging to either same table or different tables.
4. The records are populated in this table as shown in Table 2 after the domain comparison is over between them.

Table 3. The domain listing table

	TABLE_NAME	**COLUMN_NAME**	**DATA_TYPE**	**NO_DISTINCT_VALUES**	**GROUPING_NO**
1	VEMP_NAME_PHONE_MAS_FULL	EMPNO	CHAR	17888	1
2	VMOB_PHONE_MAS	EMPNO	VARCHAR2	11420	1
3	VEMP_NAME_PHONE_MAS	EMPNO	CHAR	11237	1
4	VEMP_PHONE_MAS	EMPNO	CHAR	11237	1
5	TMOB_AUTO_MSG_FRNQ	EMP_NO	CHAR	5	1
6	TMOB_CODE_EMP_MATRIX	EMPNO	CHAR	5	1
7	TMOB_STATIC_OUT_BOX	EMP_NO	CHAR	3	1
8	TMOB_CODE_MAS	CODE	VARCHAR2	23	2
9	TMOB_CODE_EMP_MATRIX	CODE	VARCHAR2	5	2
10	VMOB_INCOMING_MSGS	MESSAGE_DATE	VARCHAR2	107259	3
11	TMOB_INCOMING_MESSAGES	MESSAGE_DATE	VARCHAR2	919	3
12	VEMP_NAME_PHONE_MAS_FULL	ENAME	VARCHAR2	12927	4
13	VEMP_NAME_PHONE_MAS	ENAME	VARCHAR2	8929	4
14	VMOB_PHONE_MAS	TPNO	VARCHAR2	14989	5
15	VEMP_NAME_PHONE_MAS	TPNO	VARCHAR2	14771	5
16	VEMP_NAME_PHONE_MAS_FULL	TPNO	VARCHAR2	14771	5
17	VEMP_PHONE_MAS	TPNO	VARCHAR2	14771	5
18	VEMP_NAME_PHONE_MAS_FULL	DEPT_CD	CHAR	170	6
19	VEMP_NAME_PHONE_MAS	DEPT_CD	CHAR	155	6
20	VEMP_NAME_PHONE_MAS	ORG	CHAR	1	7
21	VEMP_NAME_PHONE_MAS_FULL	ORG	CHAR	1	7
22	VMOB_INCOMING_MSGS	MOBILE_PHONE_NO	CHAR	10258	8
23	TMOB_INCOMING_MESSAGES	MOBILE_PHONE_NO	CHAR	460	8

Notes on the In-process – Domain Listing table:

1. The purpose of this table is to hold all those field-attributes whose domain match with one another under one group.
2. The table_name, column_name and data_type are the data-table name, field-attribute name and its data type respectively. The no_of_distinct_values is the column for storing number of distinct values exist for that particular column in its table, identified with TABLE_NAME. GROUPING_NO is distinct for each group which are grouped based on the domain consistency explained earlier.

3. In any group, the column of a table whose NO_OF_DISTINCT_VALUES is the highest becomes the dimension-master table. And, it mostly contains the dimension description field also.
4. From the table-1, the column EMPNO of VEMP_NAME_PHONE_MAS_FULL is the one, which has highest number of distinct values, 17888 in grouping no.-1. Hence, this table becomes the master table for this attribute, which can be a potential dimension for a data-warehousing project.

Output Results:

Table 4. The dimension-Discovered-Table

TABLE_NAME	COLUMN_NAME	PARENT_COLUMN_NAME
VEMP_NAME_PHONE_MAS_FULL	EMPNO	DEPT_CD
TMOB_CODE_MAS	CODE	
VMOB_PHONE_MAS	TPNO	

Notes on the Output-table – Dimension-Discovered-Table :

1. All the potential dimensions as discovered are taken into this table as shown in Table 4.
2. In addition, the field that precedes the dimension discovered, in the hierarchical arrangement, and discovered using concept-hierarchy algorithm specified earlier is also captured into this table. In the table-3, first row, all the employees of a department (DEPT_CD) become the higher lever in the hierarchy of employees. For other dimensions, the hierarchy is not existing. It is a single level dimension.

Table 5. The facts-discovered-table

TABLE_NAME	COLUMN_NAME	PERIOD_DATE_COLUMN_1	PERIOD_DATE_TABLE_NAME_2	PERIOD_DATE_COLUMN_2
VEMP_NAME_PHONE_MAS_FULL	BASIC_PAY			
VEMP_NAME_PHONE_MAS_FULL	DEARNESS_ALLOW			
VEMP_NAME_PHONE_MAS_FULL		DATE_JOINED	VEMP_NAME_PHONE_MAS_FULL	DATE_SUPERANNUATION
VEMP_NAME_PHONE_MAS_FULL		DATE_JOINED	VEMP_NAME_PHONE_MAS_FULL	DATE_LATEST_PROM

Notes on the Output-table – Facts-Discovered-Table :

1. The facts as discovered are placed in this table as shown in Table 5. The COLUMN_NAME and TABLE_NAME hold the potential facts discovered.
2. Wherever the fact is a period, the COLUMN_NAME will be set to NULL, and PERIOD_DATE_COLUMN_1 is filled with one date (either starting or ending date) of the table TABLE_NAME and the PERIOD_DATE_TABLE_NAME_2 AND PERIOD_DATE_COLUMN_2 is filled with the other date.
3. The difference between these two dates results in the formation of "period" fact.

Statistical validation of the results:
For dimensions:
The retrieved data as dimensions = EMPNO, CODE, TPNO, DEPT_CD, EMPNAME, ORG
The relevant data as dimensions = EMPNO, CODE, TPNO, DEPT_CD
The statistical measures
Recall = count of intersection (relevant, retrieved)/ count of relevant = 4/4 = 100%
Precision = count of intersection (relevant, retrieved)/ count of retrieved = 4/6 = 67%

VII. CONCLUSIONS

The discovery of effective dimensions/facts from an organizational database is very much essential for any organization, which wants to implement data-warehousing systems. Using the procedures and methods delineated in this paper, even though, we may not exactly bring out the dimensions and facts exactly, but we can certainly find out the patterns and designs within the confusing milieu of databases, that might have grown across different space and time dimensions in an organization. Moreover, the probable facts/dimension fields would form part of the set of results found. No dimension/fact field would miss out. The statistical results prove a point or two in this direction. The recall of 100% and Precision of 67% are ample indication to this effect.

REFERENCES

[1] Lee, S.hyun. & Kim Mi Na, (2008) "This is my paper", ABC *Transactions on ECE*, Vol. 10, No. 5, pp120-122.

[2] Gizem, Aksahya & Ayese, Ozcan (2009) *Communications & Networks*, Network Books, ABC Publishers.

[1] Goparaju V. Ramesh, Sattiraju N. Rao and Mogalla Shashi (2011) "This is my paper" *Data-Warehousing Applications in Manufacturing Industry – Applicable Solutions and Challenges Faced* Communications in Computer and Information Science, 1, Volume 125, Advances in Computing, Communication and Control, Part 1, Pages 70-78. Available in www.springerlink.com

[2] Ponniah, Paulraj (2001) *Data Warehousing Fundamentals*. John Wiley & Sons, Inc.

[3] W.H. Inmon (2002) *Building the Data Warehouse*. John Wiley & Sons, Inc.

[4] Oracle9i Database Online Documentation for Catalogue views. http://www.oracle.com/pls/db92/homepage

[5 Oracle9i Application Developer's Guide – Coding dynamic SQL statements - http://download.oracle.com/docs/cd/B10500_01/index.htm

[6] Rahul Gupta,IIT Bombay grahul@cse.iitb.ac.in; Sunita Sarawagi, IIT Bombay, sunita@iitb.ac.in *"Domain Adaptation of Information Extraction Models"*

[7] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, Riccardo Rosati Dipartimento di Informatica e Sistemistica, Universit`a di Roma "La Sapienza" Via Salaria 113, 00198 Roma, Italy *Source Integration in DataWarehousing*

Authors

Goparaju V. Ramesh was born on 16.06.1967, and is currently working as Assistant General Manager (Information Technology) with Visakhapatnam Steel Plant. He has 14 years of experience in Information Technology, and prior to that, around 10 years in Metallurgical Quality Control aspects. He has led teams that developed many computerised systems both in conventional business systems as well as Data Warehousing & Business Intelligence Applications. Academically, he has been graduate in Metallurgical Engineering. He completed AMIIM from Indian Institute of Metals. Then, did MCA & MBA from IGNOU, New Delhi. Currently registered for Ph.D. with Acharya Nagarjuna University. He has cleared Pre Ph.D. examination. He is into English Literature. Currently pursuing MA (English Language & Literature) in School of Distance Education, Andhra University, Visakhapatnam.

Mogalla Shashi was born on 19-07-1961. She is currently Professor in Computer Science & Systems Engineering department in Andhra University College of Engineering (Autonomous), and is also Head of the department. She did B.E.(EEE), M.E.(Comp Engg), Ph.D.(Engg). From Andhra University. Her academic specialisation includes Data Warehousing & Mining, AI, Data Structures, Soft computing and Machine Learning. She co-Authored a Text Book entitled Data Structures with Program Design in "C" through Pearson Education Limited, India. She is Fellow of Institute of Engineers, Life Member of Professional Bodies like IEEE, CSI and ISTE & member of BOS in various University like Andhra University, JNTU, Kakinada, Adikavi Nannaya University, etc. She published 40 papers in journals and 24 in conferences. She guided 6 persons for their Ph.D.

KVSS Rajeswara Rao is working as Dy. General Manager (Information Technology) with Visakhapatnam Steel Plant. He has 25 years of experience in Information Technology. He is currently the Chief Information Officer of Vizag Steel. Prior to becoming CIO, he led many teams that developed many computerised systems both in business systems, process control systems, etc. He headed IT Infrastructure group also in Vizag Steel. Academically, he has graduated in Mechanical Engineering from Andhra University College of Engineering. He did subsequently, while serving in VSP, MBA from Andhra University.