Easy designing steps of a local data warehouse for possible analytical data processing

Y. Somananda Singh[1], Y. Kirani Singh[2], N. Subadani Devi[3], Y. Jayanta Singh[4]
[1]Dept. Computer Sci. & Engineering, Assam Don Bosco University
[2]CDAC, Silchar, India, [3]Manipur University, India, [4]NIELIT, Kolkata, India
ysomananda2015@gmail.com, kirani.singh@gmail.com,
Nsubadani@gmail.com, yjayanta@nielit.gov.in

Abstract: *Data warehouse (DW) are used in local or global level as per usages. Most of the DW was designed for online purposes targeting the multinational firms. Majority of local firms directly purchase such readymade DW applications for their usages. Customization, maintenance and enhancement are very costly for them. To provide fruitful e-services, the Government departments, academic Institutes, firms, Telemedicine firms etc. need a DW of themselves. Lack of electricity and internet facilities, especially in rural areas, does not motivate citizen to use the benefits of e-services. In this digital world, every local firm is interested in having their DW that may support strategic and decision making for the business.*

This study highlights the basic technical designing steps of a local DW. It gives several possible solutions that may arise during the design of the process of Extraction Transformation and Loading (ETL). It gives detail steps to develop the dimension table, fact table and loading data. Data analytics normally answers business questions and suggest future solutions.
Keywords: *A local Data warehouse, ETL process, Table-Dimension, Fact, Error event table, an outlier.*

1. Introduction

In this era of Bigdata, cloud computing and social media, there is a drastic increase in the amount of data. It is very cumbersome to store and also to process such a large data. Technically the data retrieval becomes difficult regarding time, space and efficiency. Hence the concept of multiple dimensions of the data is used to store as well as retrieve data efficiently and effectively. A multidimensional database (MDB) analyse large groups of records and gives better query performance.

A typical Data Warehouse is used to store the consolidated data from a several types of sources. DW support strategic and decision making of any business. DW can be grouped as local or global based on its usage. The local DW may to represent the data and processing at a remote site. The global DW may constitute the part of the business (logic) that is integrated across the business sites. Most of the DW system such as Clementine, Intelligent Miner, MineSet, Enterprise Miner, etc. was designed for online purposes targeting the multinational firms. The performance of the available DW system/ETL tools can be verified before using them by the users [1].

Small DW is required for most of the Government departments, academic Institutes, firms, Telemedicine etc. for better customer services. Majority of local firms directly purchase such readymade DW for their usages. Customization, maintenance and enhancement are very costly. Lack of electricity and internet facilities, especially in rural areas, does not motivate citizen to use the benefits of e-services of Government departments and institutes. In this digital world, every local firm may be interested in having its own DW that may support strategic and decision making for the business.

In practice, well defined ETL processes are required to design a workable DW system.

a] Extract:
It is the process of extracting the data from the varities of data sources. Such data source system normally stored in a different data organisation, type, or format etc.

b] Transform:
It defines a series of rules that can be applied to the extracted data using either syntactic rules or semantic rules. In syntactic rules, data are extracted from source those have different names, types. In semantic rules, different meanings of data such as daily or weekly data are extracted. Other process of cleaning, handling missing data, misspellings, error, conflicting data etc. are also executed in this phase.

c] Load:
It defines the different process of loading the data into the define Data Warehouse. The concept of a surrogate key is usually added to each row in the DW to handle problem where multiple source system may use the same key.

The data available from the real field such as census, agricultural data are not clean to process for data analytics. It is time-consuming to clean, remove the outlier or erroneous data. If not done so, it will also affect the outcomes and purposes. The options are required to remove

irrelevant and redundant data for reducing the computational cost and improving the quality of data for efficient processing tasks. An automatic error event table is necessary to collect error and fix the error or report for other possible actions [2].

Section 2 gives the background information of the study. The review of the current status of the subject in an international and national level is given in Section3. The detail of the proposed steps of designing such as all steps of ETL of a DW is given in section4.

II. Background information on the study

a. **Multidimensional Data (MD)** is organised by one or more dimensions. In other word, the MD structures are also referred to as Cubes. Oracle9i or higher version has the facility of both relational data structures, (that is tables and columns) and the multidimensional data structures (that is cubes).

b. **Dimensions are** the concepts concerning which an organisation is interested to keep its records and data fields (eg, sales1, sales2, etc). Each dimension has a database table. This associated table is known as the dimension table, which further describes a dimension. It provides rules for filtering or grouping of the data.

c. **Facts**: A MD model is an organised around a central theme that is represented by a **fact** table. A fact contains the statistical measures data. These data are quantities by which a user wishes to analyse the relationships between dimensions.

d. **Data Cube:** Modelling the data and viewed the data in the multiple dimensions are perform using the cubes. Dimensions and facts defines the cubes. In the Data Warehouse, the data cubes are n-dimensional. Some of the Multidimensional Data Models are given below.

e. **Star Schema:** The star schema is one of the modelling schema. In this, one fact table refer different number of dimension tables. A large central fact table contains a bulk of data (set of smaller dimension table).

f. **Snowflake:** The snowflake is a different version of star schema. In this some dimensional hierarchy is normalised into a set of small dimension tables, forming a shape looks like a snowflake. In this, several of dimension tables are normalised. Data in such tables are splitted into extra tables. It needs more joins to execute a query. So the system performance are normally impacted. But its advantage is that it reduces the space requires to handle the data and the number of spaces where it wishes to be updated during any data modification.

g. **Fact Constellation:** The complex business requires various fact tables to map to dimensions tables. Such schema can be viewed as a gathering of stars.

h. **ROLAP:**

Relational Online Analytic Processing (ROLAP) performs the dynamic analysis of data stored in a relational database within the database system. In a 2-tiered design, the user submits a Structured Query language (SQL) query to the database and got back the desired data. In a 3-tiered design, when a user submits a request for analysis, the ROLAP machine converts the request to SQL for submission to the database. Then, the action is performed in reverse. This is beacuse the engine converts the resultant data from SQL to a multi-dimensional layout before it is reply to users' query. The queries are created in advance. If the desired data is available, then it will be used to saves time. In other side, the query is built on the fly. As ROLAP uses a relational database, it takes more processing time to perform the tasks in MD [3].

III. Review of the current Status

A. International Status:

The older version of data models did not support many-to-many relationships between facts and dimensions. It has less built-in design facility to hold the change and time. It is unable to insert data with varying granularities [4]. Thus the extended multidimensional models were introduced which reused the multidimensional concepts of hierarchy and granularities. It also took into account the imprecision in the category of data. This model yielded a practical solution with low overheads but used Relational database technology for implementation, hence it is slow [5].

It was important to translate data into significant information in a timely and better economy manner [6]. It also gives extra time while the operational data moves into specialised analytical tools. Thus some of the models favoured to run queries straight on their operational data. Doing so, will lead to a raise in data volumes. Also, it suggests to the necessity of very fast processors or raise in the number of processors. At this time the need to scale the I/O system was also required. It is because the storage subsystem had to deal with with the raise in data capacity. The solution to this problem was to scale the I/O subsystem for capacity and to cache the database in main memory for performance. While this solution was right for small data amounts, it was challenging to implement this with large databases. Therefore extended memory concepts started coming into the picture, and Cisco's extended memory was then clubbed with oracle's RDBMS technologies to obtain the desired result.

During the last few years, some frameworks have been proposed to deal with Data warehouse propose issues [7]. Most of the frameworks so far provide partial answers that focus on two types of modelling such as multi dimensional

modelling or extraction transformation loading (ETL) modelling. Less attention has been given to unifying issues into a single structured framework and also to possible automation of the most of processes of Data warehousing. To address such problem, there is a need of a general unified and fully or semi automated process that integrates Data warehouse and steps of ETL processes. Such framework supports the model driven process. It may help the DW designer in modelling a require business decision by generating the multi dimensional model. Then it makes the physical and logical model and also generates the source code. In such study, the transformation rules are formalised using possible query or view or possible options to enable a semi automate the possible processes. A process called 'dynamic data warehousing and parallel OLAP has been proposed for optimization [8]. A design model ETL is proposed to enhance the mapping between the plan and actual process in ETL [9]. A Two-ETL phase is proposed to minimize the gap between the design and implementation of business processes [10]

Generally the performance issues arise because the queries for on line transaction processing (OLTP) systems. It accesses a miniature part of the database while OLAP may need to aggregate more major portions of a database [11]. OLAP is one of the dominant and famous technologies for knowledge discovery in a decision support system [12]. OLAP supports varities facilities of severa business applications. Generally OLAP over and over again required huge computational power. The load balancing also support in proper distributing of data or other workload amongst the participating resources to enhanced the overall performance of a DW system. The concept of load balancing was also extended to real-time OLAP systems on cloud-based architecture [13]. Proper load balancing can lead to a significant increase in the response time and throughput. Several traditional OLAP follow static data cube approach to ensure query performance [14]. Some of the available systems which are based on Hybrid OLAP are Crop Yield Prediction [15] and Data Warehouse System for Outpatient Healthcare [16].

B. National Status

In one of projects called a 'Data warehousing' executed by the government of Andhra Pradesh Government, a national agency, developed citizen's database with several facilities. It provide data and facilities related to Voters List, Food and Distribution, Industry, Professionals, Household data, Health, Economic Status and Demographic data etc to user and to the agencies. A system called eVidur was also launched for tracking social media data. It auto analyses and interprets social media big data to study the sentiment of user. It helsp in tracking several comments and other opinions in real time. It also helps to track users views on any desired facilities such a political, educational or development agendas or policy initiatives by the government. It helps the respective government or the agency to take advance measures for the welfare of the people [17]. Such a system or the e-services will be beneficial for the agencies and as well as the users.

Several MNCs (Cognizant, Infosys, and TCS etc) are developing such a DW system in India, but the clients are from other countries. Subsections from such system are used in many Indian firms. We could find some seasonal research work on this topic from many universities. Recently, a few Centre of Excellence (CoE) on Data Analytics are setting up by Ministry of Electronics & Information Technology. A per current study, lots of research projects and jobs are rising in the area of designing the data warehousing and its associated data analytics areas.

IV. Proposed Steps of Designing a DW

Several steps are going to follow for these activities. Sample setup is shown in Figure-1. Let's try to apply each of the steps to the scenario of Mini Super Market. Most of the technical steps involved in it are written in *Italic* for a user to understand clearly. Also explain the following steps in clearly understandable way.

1. Design of a primary Data warehouse
2. Design of ETL process
3. Data modelling to handle Data Quality Problems
4. Design of new rules of filtering the data
5. Demonstration a mini DW of a mini supermarket system

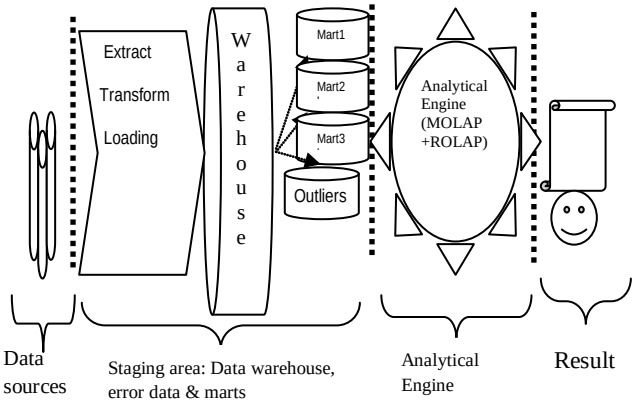

Figure-1. A sample of warehouse setup

A scenario of a Mini Supper market

There are three OLTP database sites where sales of Musical (MUSC), Electrical (ELEC) and Hardware (HRDW) products are recorded. Each site has a similar database design. Products at each location are disjoint. A product can be purchased from only one site. Product ids are assigned independently, so there is no common key. In other words, different products from different sources might have the same key. Customers are independently assigned Customer IDs at each site. Therefore the same customer may have two distinct customer ids at two different

locations. Also, different customers may have the same customer id at various sites. Customers at different locations with the same Tax File Number (TFN) are the same customer.

We need to enable the DW to answer the questions like what kind of customer is buying what kind of product, and when and how much they are spending.

V. Design of a primary Data warehouse

a. Data source, staging space, DW and marts of data:
The data are being collected from sources and maybe with different data formats. For example, some are in Oracle, some in text or in excel. The staging area is to use for processing most of the ETL steps. The error values are separated from the possible sources using different filters (examples are given below). In a data staging space, the it executes the process of extration of data and storing in a DW. Metadata stores several information such as the data source of each data item, the dates of loaded and refreshed etc. Several Data Marts may be formed based on the requirement. In this section, designing the Fact tables, Dimension tables and schema of the study model are also allowed.

b. Tables - Dimension, Fact tables and dimensional Model:

- Dimension table stores the tuple of attributes of the dimension. Here a tuple is a stored 'fact data'.
- A fact table contains the above fact data. It linked as a foreign key to dimension table. In other words, normally the 'fact data' contains possible numeric measurable or computable values or variable, that links to dimension tables.

Fact table plays vital roles in much analytical application. Let's consider a business scenario of a "mini supermarket application". This participating table can be viewed in a multidimensional ways. In the example below, the Ids of market, product, time are representing the dimensions of the possible supermarkets, products, and time respectively. In the given example 'sales amt' is a function of the given three dimensions.

Table of Fact:
Sales contains Ids (Market, Product, Time, SalesAmt)
Tables of dimensions:
Market (Market, City, State, Region)
Product (Product, Name, Category, Price)
Time (Time, Week, Month, Quarter, Year)

Sample dimensional (in possible star) model is given

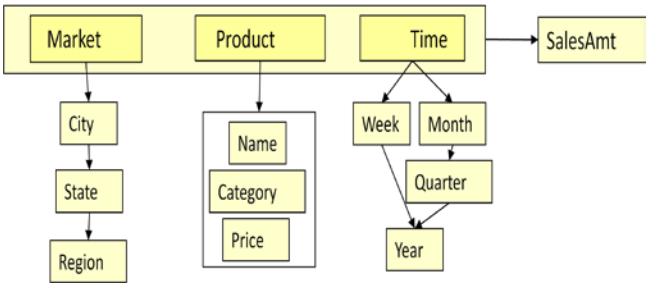

Figure2. Sample start dimensional Model

VI: Design of ETL process

a. Extract:
During this process, it extract the data from the many sources, may be from different formate too. The process called parsing is perform on the extracted data to verify the assurance of data meets as expected pattern by a business. Such design aspect can be seen from different three layers [18]. Looking in different layers makes the design easier.

1. *The source databases are integrated into a single DW.*

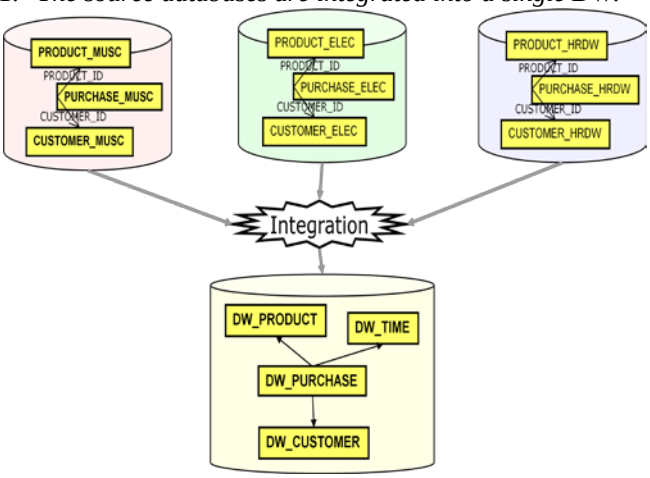

Figure3. Integration of the database in a DW

2. After the integration, before any processing start, make sure the matching of the volumes of input data by counting numbers of records in each table. If each of the three tables contains (ELEC70+HRDW60+MUSC50), there should be 180 records. Example query:
SELECT 'ELEC' "CATEGORY", COUNT(*) FROM PROD_ELEC
UNION
SELECT 'HRDW', COUNT(*) FROM PROD_HRDW
UNION
SELECT 'MUSC', COUNT(*) FROM PROD_MUSC;

b. Transform:
It deals with set of user defined rules of data on the extracted data. It can take help of syntactic rules or semantic rules. Other related process such as cleaning, handling missing values, etc is also perform.

1. The transformation may combine multiple data sources into one target. Create a target table of the same

structure. An example is given for the Product table. A similar process can be executed for the other two tables.

2. Let's add an extra column to discriminate the source, and add a surrogate key. Surrogate keys are assigned sequentially before the loading of the data. This key helps in join the dimension tables and the fact table.

CREATE TABLE DW_PRODUCT AS
*SELECT * FROM PROD_MUSC WHERE 1=0;*
ALTER TABLE DW_PRODUCT ADD (category VARCHAR2(4));
ALTER TABLE DW_PRODUCT ADD (dw_Prod NUMBER PRIMARY KEY);

This study refers some of the following queries from available e-books of Oracle Database [19, 20]. Some correction may be required as per the Database versions.

3. *The new structure of the target table (DW_PRODUCT)*

PROD ID	*NOT NULL*	*VARCHAR*
PROD NAME		*VARCHAR*
COST PRICE	*NOT NULL*	*NUMBER*
SUPPLIER		*VARCHAR*
category		*VARCHAR*
dw prod	*NOT NULL*	*NUMBER*

4. Every source product row becomes a target product row. Transformation process combines multiple data sources into one target. Let's create a sequence. This helps to generate unique integers.

CREATE SEQUENCE ASS3_DW_CUST_SEQUENCE
START WITH 1
INCREMENT BY 1;

5. Now "populate" the target table from sources by writing three queries (PROD_ELEC, PROD_HRDW, and PROD_MUSC). Example query of product ELEC is given. In the same way, carry out for the other two tables.

INSERT INTO DW_PRODUCT
(DW Prod, product_id,product_name,category, cost _price, supplier)
SELECT ProdSeq.NEXTVAL AS dw_Prod,
product_id,product_name,'ELEC' AS category type, cost price, supplier
FROM PROD_ELEC;

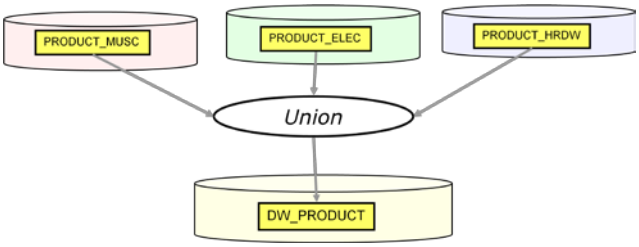

Figure4. Loading all products to a single DW product table

6. Verify the total of all records are capture or not. In our example, there should be 180 records (E70+H60+M50).

SELECT category, COUNT(*)
FROM DW_PRODUCT
GROUP BY category;

7. Transformation of customers that combines multiple data sources into one target. The same customer can appear in tables from various sources. We need a way of identifying identical customers. Two customers from different sources with the same PAN/PHn (Phone number) will be considered the same. Create a target table of the same structure. Add a column for a surrogate key. Example queries:

CREATE TABLE DW_CUSTOMER AS
SELECT city, state, postal_code, gender, PHn, occupation
FROM CUSTOMER_ELEC;

Lets ALTER the table DW_CUSTOMER by adding (dw_Cust NUMBER PRIMARY KEY);

Target Structure table (DW_CUSTOMER) looks like below

DW_CUST	*NUMBER*
CITY	*VARCHAR*
STATE	*VARCHAR*
POSTAL_CODE	*VARCHAR*
PHn	*NUMBER*
GENDER	*VARCHAR*
OCCUPATION	*VARCHAR2*

8. Merging of the customers: multiple data sources into one target

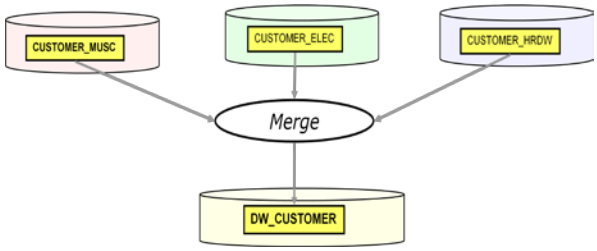

Figure5. Merging all customers to single DW customer table

Every new customer row becomes a target customer row. Every old customer row updates a target customer row. If a customer appears in more than one source, we have to decide which one has precedence. Populate the target table from sources, assuming precedence CUSTOMER_ELEC, CUST_HRDW, CUST_MUSC.

Use function MERGE to help to update if a row exists or, update it. Or if row doesn't exist, insert the data. A sample query is given for customers of ELEC. A similar process can be executed for the customers of HRDW and MUSC. If later MERGEs should update rows Inserted earlier, we can UPDATE from the WHEN MATCHED clause.

MERGE INTO DW_CUSTOMER y

```
USING CUSTOMER_ELEC x
ON (y.PHn = x.PHn)
WHEN MATCHED THEN UPDATE SET y.PHn = xPHn

WHEN NOT MATCHED THEN
INSERT (dw_cust , city, PHn, occupation,... )
VALUES (CustSeq.NEXTVAL, x.city, x.PHn, x.occupation);
```

9. Creating a Fact table of DW PURCHASE
   Assign a surrogate Foreign keys to the Fact table. Every row in the source becomes a row in the target, mapping natural, source keys to surrogate, target keys.

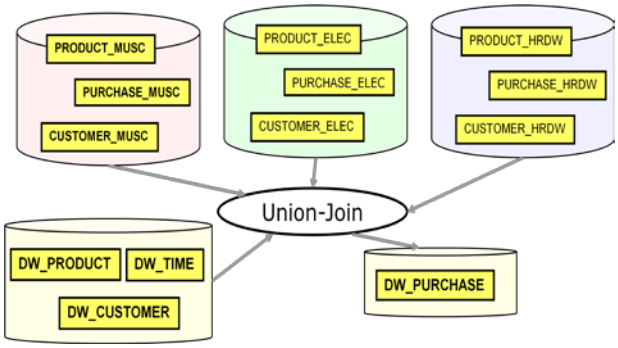

Figure6. Creating possible DW Fact table

```
Create the table using attributes.
DW_PURCHASE (
dw_Prod      NUMBER,
time_key     NUMBER,
dw_Cust      NUMBER,
cost         NUMBER(6,2),
PRIMARY KEY (dw_Prod, time_key, dw_Cust));
```

c. Loading:

It helps in loading the data into the DW. A surrogate key added to each row in the DW. This avoids a problem where multiple source systems may use the same key (e.g. customer ID).

1. Assuming the DW dimensions have already been updated from source Dimensional data, referential integrity should hold in the data warehouse. The DW PURCHASE Fact table must be populated from the source Fact table. Example

```
INSERT INTO DW_PURCHASE (dw_Prod, time_key, dw_Cust,
purchase_price)
SELECT dp.dw_Prod, TRUNC(pur.purchase_date), dc.dw_Cust,
pur.purchase_price
FROM  PURCHASE_MUSC pur, DW_PRODUCT dp,
DW_CUSTOMER dc, CUSTOMER_MUSC x
WHERE pur.customer_id = x.customer_id
AND x.PHn = dc.PHn
AND pur.product_id = dp.product_id;
```

2. Loading Sales FACT table and Sales Data Mart

Distributing a single source row into multiple target rows. Every source row gives rise to various target rows. It aims to store the data in a target table sales (prodId, custId, timeId, amountSold). Example

```
Let's use, command 'Insert all' followed by
INTO sales (prodId, custId, timeId, amountSold)
VALUES (productId, custId, weekly_start_date,   sales_sunday)
INTO Sales (prodId, custId, timeId, amountSold)
VALUES (productId, custId, weekly_start_date+1, sales_monday)
.....(plan for all sales up to saturday)
```

Data Marts can be formed for the amount sold on each weekday (Sunday to Saturday)

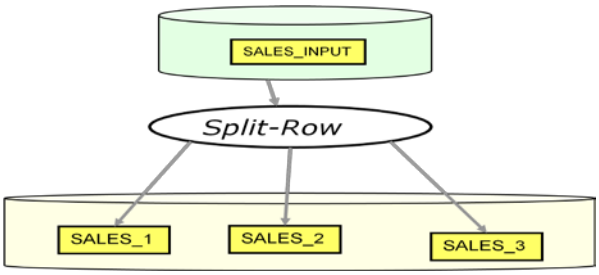

Figure7.Data mart of the amount of Sales vs weekdays

3. Data Mart of Product on Category

Suppose, we have a table PRODUCT that must be decomposed by Category. Every source row gives rise to one target row, but in different target tables. The table is split based on Category per Sales. New tables for all three Categories can be created for storing the different products categories. The process of creating one table and the process of loading of all categories are provided.

Create table PROD_ELEC using he fields as productId, product_name, cost_price, supplier, same structure as PRODUCT;

```
INSERT FIRST                    /*Only the first successful record
WHEN category = 'ELEC'  THEN
     INTO PROD_ELEC
    VALUES (productId, product_name, cost_price,supplier)
WHEN category = 'HRDW'  THEN
    INTO PROD_HRDW
   VALUES (productId, product_name,cost_price,supplier)
ELSE     INTO PROD_MUSC
   VALUES (productId,product_name,cost_price,supplier)
SELECT product_id,product_name, categoryName,
cost_price,supplier
  FROM PRODUCT;
```

Now, the table of DW (means the three dimension tables) and one DW fact (DW_PURCHASE) are loaded with records from the individual table. The schema between the tables is Star Schema. After these processes, verify that we have not lost any rows-(ELEC70+HRDW60+MUSC50), a total of 180 records.

VII: Basic Data modelling for Data Quality Problems

Some of the common data problems are provided.

- Multiple identifiers: Data from different sources will use different keys.
- Multiple field names: Data from a different source will use different field names
- Different units: Different units of measurement
- Missing values and Orphaned values
- Developing related model/data marts as a Business requirement

Table1: A sample data.

PROD_ID	NAME	CATEGORY	PRICE
1	Soda Water	Drink	0.82
2	Bok Choy	Vegetable	2
3	Black Olives	Deli	4.25
6	Birdseye Chili	Vegetable	9
9	Horehound	Drink	1.8
10	Soda Water	Drink	1.5

Some possible solutions are provided below.

- *Display any names that are duplicated*

```
SELECT name from PRODUCT
GROUP BY name HAVING COUNT(*) > 1;
```

Output: Soda Water

Data analysis requires unique PRODUCT names. Duplicate data can be recorded in an Error event table for future processing.

- Checking for outliers/unusual values

1. Create functions that calculate reasonable lower and upper bounds on PURCHASE_PRICE. Example

```
CREATE OR REPLACE FUNCTION minReasonablePrice
        RETURN NUMBER IS
        val NUMBER;
        BEGIN
        SELECT AVG(PURCHASE_PRICE)-
        3*STDDEV(PURCHASE_PRICE)
        INTO val
        FROM PURCHASES;
        RETURN val;
END;  /*Create another function for a maxReasonablePrice
```

2. Then, report all values outside the reasonable range.

```
SELECT 'PURCHASES', 10, SYSDATE, 3
FROM PURCHASES
WHERE PURCHASE_PRICE NOT BETWEEN
minReasonablePrice AND maxReasonablePrice;
```

- Data cleaning: Validating Foreign keys in the Fact table

The Fact table lies in the centre of a Star Schema to linked to Dimension tables by foreign keys. These foreign key values must link to surrogate primary keys in Dimension tables. When a source Fact table is received, it has natural foreign keys. These must be converted to surrogate keys. Let's apply on a source table PURCHASES to report any violations of natural FKs.

```
SELECT 'PURCHASES_MUSC', 12, SYSDATE, 3
FROM PURCHASES_MUSC p
WHERE p.product_id NOT IN (SELECT product_id
from PROD_MUSC);
```

In the next part of the study aims to explore the design of new rules of filtering the data

a. Create an Error Event Table to capture the error or outlier data.
b. Design a set of the new rules for filtering the data.
c. Form new rules of modelling, data cube, slicing, dicing, etc.
d. Design algorithms for speedy operations. For example, Probabilistic clustering for forming possible data marts [21]. An automatize warehousing process (less human intervention) as to fit space of cubes and data marts etc. It will help to store the data efficiently. Consider usages of suggestive tips in case of performance issues in case of a large aggregation of data [22]. Speedy retrieval of only the required data from bulk data. Advance options such as the novel frequent pattern tree allows mining of various items during development of DW[23].

VIII: Conclusion

Most of the DW was designed for online purposes targeting the multinational firms. This study provided the necessary designing steps of a local data warehouse for possible analytical data processing. The technical steps of ETL are given in detail. It gives detail steps to develop the dimension table, fact table and loading data. It highlights several step by step solutions to extract data from several sources. Also, several possible options of Transformation of data in each different step of a DW are considered. The studies provide some hints on the environment of DW, which may enable the Data analytics for answering the business questions of any firm.

IX: References

[1] TA Majchrzak, T Jansen, H Kuchen, "Efficiency evaluation of open source ETL tools." *Proceedings of the 2011 ACM Symposium on Applied Computing*. ACM, 2011.

[2] R Kimball, J Caserta, The data warehouse ETL toolkit: practical techniques for extracting, cleaning, conforming, and delivering data. John Wiley & Sons, 2011.

[3] TB Pedersen, CS Jensen, "Multidimensional database technology." *Computer Journal*, pp.40-46,2001

[4] TB Pedersen, CS Jensen, CE Dyreson, "A Foundation for capturing and Querying complex multidimensional data"; *Information systems*, Elsevier, July 2001, pp 383-423

[5] D Feinberg, MA Beyer, "Magic quadrant for data warehouse database management systems." *Gartner Research Note*, 2008.

[6] M. Poess, R. Nambiar, ”Building Enterprise Class Real-Time Energy Efficient DSS” in *Enabling Real-Time Business Intelligence*, Vol 84, pp 36-5, 2011.
[7] F Atigui, F Ravat, R Tournier, G Zurfluh, "A Unified Model-Driven Methodology for Data Warehouses and ETL Design." In *ICEIS*, pp. 247-252. 2011.
[8] Q Chen, M Hsu, U Dayal, 'A data-warehouse/OLAP framework for scalable telecommunication tandem traffic analysis', *Proceedings of 16th International Conference on Data Engineering (Cat. No. 00CB37073)*. IEEE, 2000.
[9] SHA El-Sappagh, AMA Hendawi, AH Bastawissy, "A proposed model for data warehouse ETL processes." *Journal of King Saud University-Computer and Information Sciences 23.2* (2011): 91-104.
[10] A Nabli, S Bouaziz, R Yangui, F Gargouri" Two-ETL phases for data warehouse creation: Design and implementation." *East European Conf. on Advances in Databases and Information Systems. Springer, Cham, 2015*.
[11] F Dehne, Q Kong, A Rau-Chaplin, H Zaboli, "Scalable real-time OLAP on cloud architectures." *Journal of Parallel and Distributed Computing* 79, pp.31-41, May2015
[12] H.Zaboli, “Parallel OLAP on Multi/Many Core and Cloud Platforms”, *PhD thesis*, March 2014.
[13] J.Caskey, "Load balancing strategies for Cloud-based Real-Time OLAP", *Project report*, April 2013
[14] F.Dehne, H. Zaboli, ”Parallel Real-Time OLAP on Multicore Processors”, Proc. *2012th IEEE/ACM*, 2012.
[15] VM Ngo, NA Le-Khac, M Kechadi, "An Efficient Data Warehouse for Crop Yield Prediction", in *proc.14th Int. Conf. on Precision Agriculture*. June 24-27, 2018
[16] TMJ Al Taleb, S Hasan, YY Mahdi, "Data Warehouse System for Outpatient Healthcare", *Journal of Fundamental and Applied Sciences* 10, pp.187-192, 2018
[17] *www.cdac.in, [accessed on 10 Oct2018]*
[18] N. Subadani, L.Prabhakar, "A Data Warehouse system for Human Resource Management in a Distributed Software Development" *ADBU Journal of Engineering Technology*, 5(2), 2016.
[19] Susan Hillson, Lilian Hobbs, Shilpa Lawande, E-Book: Oracle 10g Data Warehousing, 2004
[20] Gavin Powell, E-Book: Oracle Data Warehouse Tuning For 10g, 2015
[21] S McClean, B Scotney, P Morrow, K Greer, McClean, "Knowledge discovery by probabilistic clustering of distributed databases." *Data & Knowledge Engineering* 54, no.2, pp.189-210, 2005
[22] B. S. Zaman, B. Kumar, Z. Azim, Y. J Singh, "Suggestive Local Engine for SQL Developer: SLED." *ADBU Journal of Engineering Technology* 4, 2016.
[23] J Han, J Pei, Y Yin, "Mining frequent patterns without candidate generation." *In ACM sigmod record*, 29, no. 2, pp. 1-12. ACM, 2000.

Author(s) Profile

Yumnam Somananda Singh is working in Faculty of Computer Science of The South East Manipur College, Manipur India. He has master and M.Phil in Computer Science. Earlier he was with in Faculty of Computer Application, Institute of Management Studies & I.T (IMSIT) Aurangabad, Maharashtra, India for 5years. Now is a Ph.D scholar at Assam Don Bosco University. His areas of interest are Distributed Database, Data warehousing, Data Mining, ETL, Data Science etc.

Yumnam Kirani Singh has completed a Master's Degree in Electronics Science from Guwahati University in 1997 and got Ph. D. degree from Indian Statistical Institute, Kolkata in 2006. Served as a lecturer in Electronics in Shri Shankaracharaya College of Engineering & Technology from Jan 2005 to May 2006. Joined CDAC Kolkata in May 2006 and worked there before coming to CDAC Silchar, in March 2014. Developed Bino's Model of Multiplication, ISITRA, YKSK Transforms and several other image binarization and edge detection techniques. Interested in working in the application and research areas of Signal Processing, Image Processing, Pattern Recognition and Information Security. Also published several papers in national and international journals and conferences.

Ningombam Subadani Devi holds a Ph.D from Manipur Institute of Management Studies (MIMS), Manipur University, India. She also holds MBA specialization in Information Technology from Sikkim Manipal University. Her interest areas are Software Engineering, Management Information Systems, Database Management System, Data Analytics and Data Science.

Yumnam Jayanta Singh is working as Director at NIELIT, Kolkata. He received his PhD from Dr B.A Marathwada University in 2004. He has worked with Don Bosco University (IN), Swinburne University of Technology (AUS), Misurata University, Keane (India & Canada), TechMahindra, Skyline University College (UAE) etc. His research areas are Blockchain, Data Science, ETL, Data Warehouse and Mining, Real-time Database system, and Image processing. He has produced several papers in International and National Journals and Conferences.