Abstract— Data warehouse can be defined as the collection of data that includes entire data of an organization. It came into existence due to more focus of senior management on data as well as data driven decision making and migration of the data from the legacy system to the new system. Historical collection or copy of OLTP data for analysis and forecasting in order to support management decision. It is realized by the organizations that for expanding their business as well for making decisions more effective, they need to analyse the data. Organization decisions depend on the data warehouse so the data should be of utmost quality. For making the decisions more accurate for the organization, it is mandatory that the testing should be done very efficiently to avoid wrong data pumped into the database since it can create ambiguity for senior management to take correct decisions. This paper covers various approaches which should be followed in different scenarios of testing phases. Our experimental research on these approaches show increased productivity and accurate flowing of the data into the final warehouse. If these approaches are followed accurately, then tester can assure that the data pumped into the warehouse is accurate.

INTRODUCTION

Relevance and importance to industry:-

Since Data warehouse testing revolves around the validation of data, it is the one of the most complex as well as challenging task in the field of testing. It includes the validation of the data which is going into the final data warehouse. It consists of testing the complex business rules and transformations. If defects are not identified in the early phases, it will be very costly to fix it in later phase. It includes the validation of the data which is going in the final database (which is called warehouse). If the data validation is not done properly, then incorrect data will be pumped in the final warehouse which can affect the business drastically.

Few guidelines for data warehouse tester:

All the transformations should be checked compulsorily because if any of the steps is left leading to incorrect data in next steps and finally leading to populate warehouse with improper data.

The important thing here is that the complete logic of the transformation should be checked and testing time should vary based on the complexity of the logic involved. There are many business processes that have very low tolerances for error, e.g. financial services.

Business users should be part of data warehouse testing because they are the reason for existence of data warehouse and they are the final user of developed system.

Extract, Transform and Load (ETL) testing should be a priority, accounting for as much as 60% of testing.

TECHNICAL WORK PREPARATION

The data warehouse can broadly be segregated into three parts:

1. Extraction, Transformation and Load (ETL)
2. Reports
3. File Testing and Database testing

Extraction, Transformation and Load (ETL)

At higher level ETL can be defined as picking data from different databases and dumping the data in the common database after applying transformation as directed by the business users.

- Extraction:- It can be defined as extracting the data from numerous heterogeneous systems.
• **Transformation:** Applying the business logics as specified by the business on the data derived from sources.

• **Load:** Pumping the data into the final warehouse after completing the above two process.

There are different ways of testing the ETL process. Remember, **ETL testing is very technical as well as very time consuming.** The process should be performed very carefully. If even one business rule is not tested properly, the data which will be pumped in the DWH will have wrong data and the interpretation of the result will be different as well as it may affect other business rules also.

1. **ETL testing:** The ETL testing can be broadly classified into 2 different scenarios:

   a. **Data migration from the legacy system into the new system:**

   ![Diagram showing data migration](image)

   **Challenge:** There is no proper documentation for the legacy system, team has to analyze the legacy system and then work for building the new system.

   **Conflicts:** If all business rules and transformation are not captured accurately, then the testing team cannot assure that the new system is the replica of the legacy system.

   **Resolution:** Testers have to feed the same data into the old and new system and compare the results. It is kind of black box testing in which there is no need for the tester to know the system. If there is the difference in the outputs then raise the bug. This approach is very robust for testing the data migration projects.

   **Example:** Data migration from the tandem server to DB2 server.

   **Explanation:** The tandem server is the old server, all the information from the source system is send to the Tandem server. After the data is transferred to the Tandem server, the transformation logic is applied and then the data is send to the Target Tables.

   **Scenario:** The management has decided that the Tandem server has to be shutdown/removed and all the transformation which is applied in the tandem server have to be implemented in the DB2 server. Basically, the business wants that all the transformation should be done and finally the data should be loaded in the final table same which was followed in the Tandem Server.

   **Design of the system:** As a tester, we don’t have to be concerned about the design but basically we have to see that old and new system is populating the duplicate data.

   **Testing Approach:** Now here the challenge is encountered, as a tester we have to check whether the same data is populated in the final tables. But there is only one set of final tables. Then how to verify that both the processes give the same data? Well there is solution for this challenge, suppose there are 20 final tables in which the data is finally loaded. Now create the same 20 tables in another schema (let the schema name be TEMP01) with the same structures as compared to the final tables.

   After creating the TEMP01 schema, run the old job, when the old job is run, it will populate the final tables. Take whole of the data from the final tables and dump the data into the TEMP01 schema.

   After the old process, run the new process and populate the data in the final tables.

   After performing both processes, the data is finally loaded and the tables are ready for comparisons.

   The comparison between the two schemas is very simple, we just have to apply the minus operator (in oracle) or except operator (in DB2).

   Suppose there is a final table called employee. We have to compare TEMP01 schema table employee and the final table employee. We can use the following command.

   ```sql
   Select * from final. Employee
   Minus
   Select * from temp.employee
   ```

   After executing the above command, the difference in the tables will be highlighted. If there is no difference then the data is populated correctly else report an issue.

   **Note:** For making the testing more accurate, follow the same command vice versa also, like minus the a
table from a to b and then minus the tables b from a. This way tester will know whether there are extra rows present in either of the two tables.

Apart from that, testers can also mock up the data in the source tables. For example transformation says ‘Location of employees with employee id ranging between 2000 and 5000 should be converted to Bangalore in target table, else direct map both the columns’.

Data mock-up:- Mock the data in the source table for couple of records as any other location for employees ranging between 2000 to 5000 and check in target table whether it is populating correctly or not.

b. New Implementations:-

This approach is followed when the data warehouse is newly implemented, in the sense that new business rules are getting implemented. In this approach, testers have to test the warehouse from Source to Target. There will be document called ‘Source to target mapping’ document will be provided for writing the test cases as well as for testing.

Challenge: - There are millions of records as well as the complex transformations which will be loaded into the final warehouse. It will be impossible for the tester to check the data row by row and ensure that the data is fine.

Conflicts: - If all the transformations are not done accurately then it is not possible for tester to certify that the data fed is correct since millions of records will be pumped in the warehouse.

Resolution: - There are 2 ways for testing.
1) Sampling of data:- This is the most common technique followed; tester can pick the couple of records and perform sampling to ensure that all the business rules are covered properly.

Example1:- Suppose there is employee table, the transformation is

Example 2:- Suppose there is employee table, the transformation is

2) Database validation by writing the query:- Write the sql query in such a way that that can test the whole table. The queries can be written effectively by using the sql functions. The second option is the best option as it assures that the data is correct. I will be taking the above two examples for proper comparisons of the data.

Example1:-Suppose there is employee table, the transformation is

Example 2:- Suppose there is employee table, the transformation is

LOCATION OF EMPLOYEES WITH EMPLOYEE ID RANGING BETWEEN 2000 AND 5000 SHOULD BE CONVERTED TO BANGALORE IN TARGET TABLE, ELSE DIRECT MAP BOTH THE COLUMNS.

In the scenario given above, tester can pick 30 records for the employee id ranging between 2000 and 5000 and verify whether the data is populated properly. If the records are populated properly then transformation is done accurately.

Example 2:- Suppose there is employee table, the transformation is

LOCATION OF EMPLOYEES SHOULD BE CONVERTED TO THE FIRST LETTER OF THE LOCATION AND HAVE TO BE POPULATED IN THE TARGET TABLE, LIKE BANGALORE CONVERTED TO ‘B’, MUMBAI CONVERTED TO ‘M’

In the above scenario, pick the set of rows for different locations and do the sampling on the target database.
By writing the above query, the data warehouse can be tested end to end ensuring that the data pumped is correct.

**NOTE:** The data base query can vary in different databases. I am giving only the approach of testing the entire system.

2. **Reports testing**

The Reports are the end result for which the whole data warehouse is build. The reports can be defined as “Pictorial or tabular representation of the data stored in data warehouse. Basically for the end users, it is used for taking the decisions.” It is the report which senior management looks into for taking the decisions.

There are different ways to test the reports, the different approaches are:

a. **Report Migration from the old system to the new system:** It is possible in the scenario of data migration from one tool to another. Like if I take an example of migrating the report from cognos server to the SSRS server, in this example the functionality of the report is not altered but due to cost issue or performance issue the report have to be migrated from one tool to another.

**Challenge:** There is no proper documentation for the legacy system, team has to analyze the legacy system and then work for building the new system.

**Conflicts:** If all the business rules and transformation are not captured accurately, then the testing team cannot assure that the new system is the replica of the legacy system.

**Resolution:** The testers have to feed the same data into the old and new system and compare the results. It is kind of black box testing in which there is no need for the tester to know the system. If there is the difference in the outputs then raise the bug. This approach is very robust for testing the report migration projects.

**Example:** Report migration from Cognos server to the SSRS server.

**Explanation:** The Cognos server is the old server, all the reports were generated in this tool server. Now business has decided that due to cost issue, all the reports have to be migrated from the cognos to SSRS server.

**Scenario:** The management has decided that due to cost issue, the report have to be migrated from cognos to SSRS server along with the additional features.

**Design of the system:** As a tester, we don’t have to be concerned about the design but basically we have to see that old and new system is populating the duplicate data.

**Testing Approach:** Now here the challenge is encountered, as a tester we have to check whether the same report is getting generated in both the systems. In order to test, report in both the systems have to be generated with the same parameters.

There is report called “Employee Report”, This report contains the following columns:-

1) Employee ID
2) Employee Name
3) Department name
4) Salary

The parameters for generating the report are:-

1) Department Name (Multiple select option)
2) Region (Drop Down)

The tester will generate the report based on some criteria. After generating the report, tester can check the following criteria:-

1) Check the layout of the reports.
2) Check the count of records in both the reports.
3) Take 20 records for sampling purpose and verify in both the reports.

If any one of the criteria is not matching, tester can log the defect. After all the above criteria is matched, tester can export the reports in excel sheet and can compare each row.
**b. New Implementations:**

1. **Layout Validation:** Layout validation is probably the easiest to test in reports. It surrounds on checking the font size, number of columns, color of the report or the specific column, etc. I will take an example to explain the same.

Suppose the RS given to you states as follows:
1. The report should contain the column Empid, Empname, deptname, location name.
2. The number of pages is shown at the bottom of the page.
3. The report should be generated in pdf format and can be derived into .pdf, .ms excel, .word, .pie chart format.
4. The title of the report should be “Employee Information”.

These are some list of requirements which I have taken as an example.

For testing the above requirements, the tester have to open the report and have to check the reports but prior to testing the tester have to write the test scenario as well as the test cases.

2. **Data Validation:** Data validation is main in the report validation. Data validation basically deals in sampling of reports after writing the SQL queries. The query is written according to the condition which is present in the RS.

I will give an example to explain the same.

The report is generated for the condition that the report should contain the column Empid, Empname, deptname, location name.

For the above query the reported is generated and the data is displayed. For validating the data present in the report is in sync with the data in database, you have to write the SQL query.

The SQL query is given as below:

```sql
Select e.empid, e.empname, d.deptname, l.locationname
From employee e
Join department d
On e.dept_id=d.dept_id
Join location l
On l.location_id=d.location_id.
And e.emp_id=123;
```

This above query will solve the problem of validating the data.

Similarly the Stored Procedure can also be written so make the validation easier and efficient.

**Data validation is also known as Database testing.**

3. **Calculation Validation:** Calculation validation is basically checking the formula which has to be applied for different column. The formula is generally tested in the stored procedure or where the rules are defined.

To explain the above validation, the example is as follows:

The formula for

\[
\text{Average} = \frac{\text{sum of elements}}{\text{number of elements}}
\]

Suppose for any specific column, the above formula have to be implemented, take an example for finding the average salary for each department.

In the above scenario, the tester should go in the stored procedure or the rules and manually check whether the same formula is being applied or not.

This is the second approach which basically consists of the 3 validations. This approach is generally used when the report is developed from the scratch.

**Testing of Files with database and testing between the two files:**

File testing is extremely complex when it is done manually. It is more prone to errors as well as cannot be appropriate. Normally tester involves them by comparing the two files manually row by row and column by column.

Instead of performing the operation manually, we can use the MS-Access. MS-access is the free database given by Microsoft and is available with windows OS. The tester can import the files in the MS-Access and can run the database queries to validate the data in both the files.

**Example1:** There are source and target file called employee which have to be tested. The source is given as file and the target is also the file, if we test manually then it will take time
taking as well as cumbersome job. In order to avoid the same, tester should follow the steps given below:

1. FTP the file from the UNIX box and place in the local system.
2. Import the files in the MS-Access (The procedure for creating the database and importing it into the MS-Access will be told during the presentation.).
3. After importing the files, fire the sql query in the same way we fire in databases.

Scenario 1:- The employee between the 1000 and 2000 should have the location ‘Bangalore’.
For comparing the same, if we do it manually then we might have done sampling, but by sampling we cannot assure that data is fed correctly. But if we follow our process, we can fire the sql query in both the files.

```
Select distinct (location) from source. Employee where employeeid between (1000 and 2000)
```

```
Select distinct (location) from target. Employee where employeeid between (1000 and 2000)
```

Example2:- The comparison has to be done between the file and database. The source given is file and the target is database. The file called employee has to be directly exported from the file to the data base.

Now suppose the file contains the millions of records which are populating in the database, it is very difficult for the tester to test each and every row.

So in order to avoid the problem we can import the file to MS-Access and the can fire the same query which we fire in the data base.

In this way we can perform the effective testing of the data warehouse which is error free and very effective.

**ACKNOWLEDGMENT**

**Author Vibhor Raman** gratefully acknowledges the contributions of his Father Shri Govind Kumar for consistently motivating him to write the paper in high quality and believing in author that he can write the paper.

**REFERENCE**

Title:-www.google.com