Data Generation and Report Testing (End-to-End) Automation of Retail Insights

Submitted By

Farzan Cooper

15MCEC07

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Data Generation and Report Testing (End-to-End) Automation of Retail Insights

Major Project

Submitted in fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Farzan Cooper

15MCEC07

Guided By

Prof. Vipul Chudasama

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Certificate

This is to certify that the major project entitled **"Data Generation and Report Testing (End-to-End) Automation of Retail Insights"** submitted by **Farzan Cooper (Roll No: 15MCEC07)**, towards the fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, hasn't been submitted to any other university or institution for award of any degree or diploma.

Prof. Vipul Chudasama
Guide & Associate Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Farzan Cooper**, Roll. No. **15MCEC07**, give undertaking that the Major Project entitled "**Data Generation and Report Testing (End-to-End) Automation of Retail Insights**" submitted by me, towards the fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. Vipul Chudasama

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Vipul Chudasama**, Assistant Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation she has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Farzan Cooper

15MCEC07

Abstract

With Rapid Advancement on more and more new technologies, no-one wants to do manual work when it could be automated or made fast to execute. The Automation for Retail Insights Testing included me making fault-correction and updates to the existing system via learning and using OpenScript. OpenScript is a scripting platform for generating automated test scripts using Java as its base language. Merging an intuitive GUI with the robust Java language, OpenScript aids requirements ranging from novice testers to advanced QA automation.

In this paper, I am going to demonstrate what Automated Testing for a System means, why and how to approach that system, and what can one achieve in this automated system over the manual testing of system. I have pointed out several approaches to automated testing, and how much proportion of a system testing should be automated. Because automating a system 100% is not only impractical, but inefficient as well. I have explored the possibility and challenges of current system automation and the Future Scope.

Abbreviations

QA	Quality Assurance
RI	Oracle Retail Insights.
RMS	Retail Merchandising System
RDE	Retail Data Extractor
OBIEE	Oracle Business Intelligence Enterprise Edition
BI	Business Intelligence
SDE	Source Dependent Extracts
ETL	Extract, Transform and Load
ODI	Oracle Data Integrator
SGD	Secure Global Desktop
ETL	Extract, Transform and Load
APS	Automatable Portion of System
PA	Progress of Automation
SSH	Secure Shell
Rlogin	Remote Login
WinSCP	Windows Secure Copy
FTP	File Transfer Protocol

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 General Idea about OpenScript

OpenScript is built on a standards-based platform and provides the foundation for OpenScript Modules and Application Programming Interfaces (APIs). OpenScript APIs are used to build scripts for testing Web applications. The OpenScript API consists of a set of procedures that can be used to customize the scripts within the development environment. The API can also be used by advanced technical users to enhance scripts for unique testing needs.

OpenScript is the next generation environment for developing Oracle Application Testing Suite scripts for Web application testing. OpenScript provides the following a platform for Eclipse based Scripting, to test modules, Tree View Structure of Files, inbuilt data-bank, etc.[1]

1.2 General Idea about Retail Insights

Oracle Retail Insights delivers mobile-enabled insights for retailers who want to leverage data-driven insights powered by retail science to improve business performance. Oracle Retail Insights is comprised of Merchandise insights and customer insights.

Oracle Retail Insights supports customizable presentation and configurable data management, and employs a complete, open, and integrated Oracle technology stack, from storage-to-scorecard.[2]

1.3 Installation of OpenScript

The first thing that i did as soon as I was assigned assignments was to study Openscript, and for that i needed to install the Oracle Application Testing Suite. The installation is Quite easy and it takes time varying from 5 minutes to 45 minutes based on your required components. I only needed Openscript, so my installation was done in about 10 minutes. After which, we had a week of training about how to Record an activity being carried out on Browser, on Oracle EBS forms, etc.

1.4 In Depth about the Assignment

The first thing i was told to do was to create a project to automate metric testing which are developed internally for OBIEE product, which i created using OpenScript. The thing in this project is that we were to test some of the developed Metrics for a particular company.

If you have to test about 700 metrics to be tested manually, it would take a lot of human effort and a whole lot of time. Particularly, i had to automate that testing. Now the previously existing metrics were already manually tested, and we knew the way in which it was being tested. So, i recorded the manual tests of some of these metrics in Openscript along with different dependency metrics which could help me in editing it to automate for all kinds of metrics. When i started developing this project, i divided the project into certain parts like Database Access part, Data Generation Part, etc. which I've mentioned in detail in section 5.3. But the issue here was something different. The metrics depend on some other attributes, like item number, fiscal date, etc. Some metrics have to be tested on Date level, whereas some have to be tested at week level, period level and so on. So i created a consolidated format for defining these metrics along with these dependencies. Yes, that is a manual process for sure, but as i already mentioned in abstract, that a 100% automation is not only impractical, but inefficient as well. The metrics would work fine, but for the older metrics which were tested. New metrics were taking data from new databases, new tables. Also many had anew parameters against which it was to be tested.

1.4.1 Beginning of the Assignment

When the testing started, it turned out that the automation itself was taking more time compared to what it should take ideally (Still, it was lesser than the manual task). A deep peek gave the insight that the original code was working through abbreviations for parameters from one file, then mapping them via another file. If this dependency could be removed, a subtle amount of time could be saved.

So the Scripts were edited accordingly, and File dependency removed. This saved a significant amount of time that was being wasted before. Metrics which passed, were being marked as so. The ones which failed, were debugged as to what is causing them to fail. Most of them were failing due to unavailability of data, Dependency on another failed metric, or due to dependency on multiple parameters for its value.

Within the time being, i also made a brief Flow about what function is being called from where. This allowed us to see where the loopholes were being formed, that means that if one function is failing, we could easily take a peek that which more functions are dependent on it which would also fail. This could help during the debugging of the failed metrics.

Chapter 2

Literature Survey

Data Representation and Knowledge making from that Data is a lot more important in today's world of increasing information. A better way to do it is to store each dimensions in as a particular value, in a table and make sense out of it. But the reports generated with it should also be correct. Here, we'll see why we are doing automation for the OBIEE system testing, and how it will affect the workload and time commitment.

2.1 Motivation

Beforehand, the metrics that were being developed for Retail Insights, were developed and tested manually, so that means more resources are occupied on this for more time period. That makes testing other stuff to get delayed. Of course, this time increases linearly with number of metrics which is something none would want. This bound time and these bound resources could be used elsewhere if it did not take this much amount of time.

Before, when manual testing was being done, it took even weeks to get the results tested and working. By automating all that stuff, we can make the process faster and give out results in days. Of course, there would be no automated system without a manual test, and debugging would be manual, but still it would decrease time consumption. And once we automate this process, we could try and sort out other manual processes to also automate if possible.

Once a manual test is done, we understand how these processes can be automated, whether it should be automated, and is it worth automating. All these questions can

be answered by figuring out how much is that process repeated. If it is repeated much number of times, it can and should be automated.

2.2 Importance of Automation

First of all, you make a system only when it is beneficial to you, and not otherwise. But the main question arises about how does one decide/determine the benefits of a system especially when it is not even made. But it is simpler to answer than you can think. You can determine at first, the system you are currently using, is it working fine? Can it be made better? If yes, how much better? In what context? i.e. Time Saving, or Space Storing, how can the new system be more convenient than the existing one? Once these questions are answered, you can know whether you want to make that system or not.

Similar is the case with automation. You must know what portion of your manual system is repeatable, and how many times is it being repeated? Is it enough to waste more time of yours, which can be utilized elsewhere? If yes, you need automation over there.

At the very beginning, we should evaluate the existing system. As discussed previously, the manual testing for many things, mainly the metric testing is tedious and time consuming, which in turn also occupies the resources. If there could be a particular system that can do this work faster and in a smarter way, with the least amount of resources, it should be created. So, now we need to determine whether the system which will be created will be worth the effort in future.

Determining whether the system will be of use driver us to determining how much portion of a system can be automatable. This depends on a few factors, such as how many of the total test cases are repetitive, and thus, automatable. This can be called as Automatable Portion of System as in Equation 2.1.

$$APS\% = \frac{no.of test cases that can be automated}{total number of test cases} * 100 \tag{2.1}$$

We decided that this Ratio should be higher that 70% to say that we need this system badly. So, I took a sample of 712 test cases. 607 out of which were easily automatable. So that gives out APS at 85.25%.

But, I also needed to keep track about how much of work is done, and I used to track it by counting the Progress of Automation as shown in Equation 2.2.

$$PA\% = \frac{no.of test cases that have been automated}{no.of test cases that can be automated} * 100 \tag{2.2}$$

Chapter 3

Technologies

3.1 OpenScript

- We can import an entire Library, or we could just import the rules defined in some particular Library.
- Integrated Tools describe that once developed System can easily be used by Oracle Test Manager, and if the scripts are for load testing, by Oracle Load Test tool as well.
- Playback of the actions recorded. You can also edit, which is usually needed to suit your needs of Automation.
- You can perform error recovery through various options, like you can set a breakpoint, or print console messages, etc. to track your fault if any.
- OpenScript not only records Browser Activities, but also Java applications activities, as Oracle e-Business Suite is a java based application basically.
- For our access, OpenScript provides us with Databanks so that we can store our data when needed without connecting to outside database sources. However, support for outside database source connection are also available.
- There are File Acccess Functions given which can be used as our requirements. We are already using these functions in FileAccessLibrary.
- Some more features involve Adobe Flex Test module, Oracle Fusion/ADF Test Module, the Seibel Test Module and Shared Data Module.

Fig. 3.1 shows how an OpenScript Environment looks like.

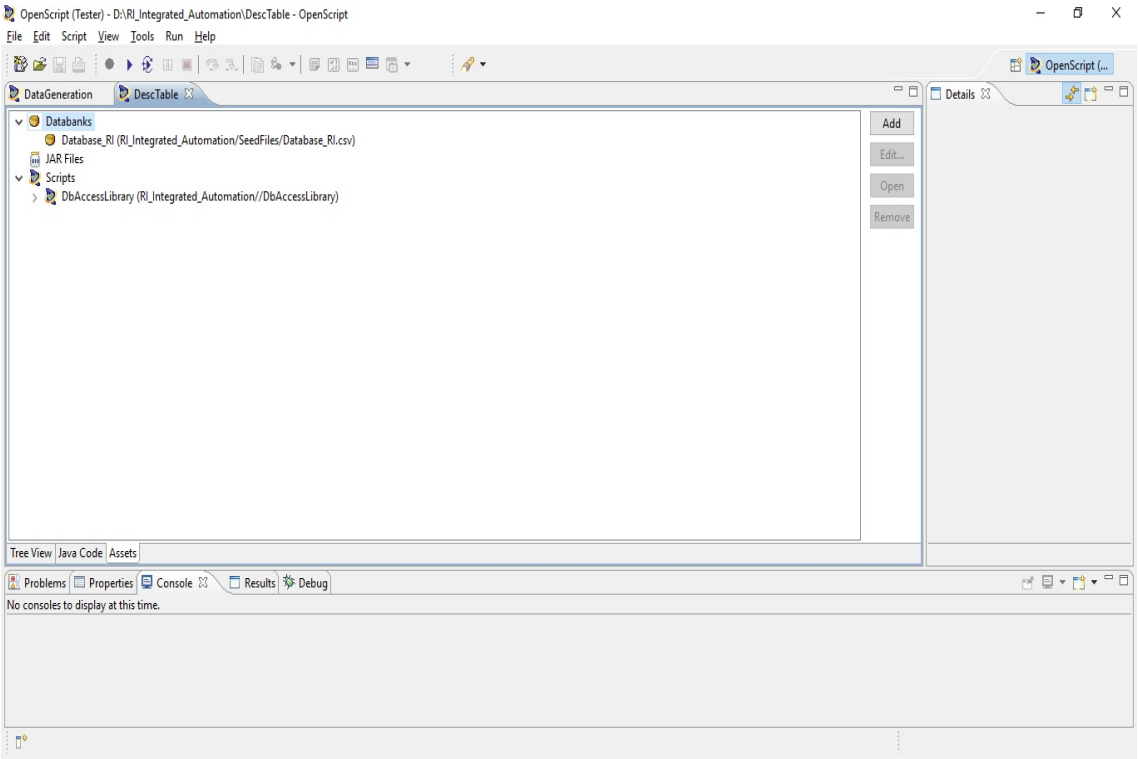

Figure 3.1: OpenScript Environment

3.2 Oracle Retail Insights

I am listing out some features which makes this product better than others.

- The Oracle Retail Insights data model is based on Oracle Business Intelligence (BI) applications and is extended to support a retailer's business intelligence data needs. It is designed to maximize performance with Oracle Data Integrator and Business Intelligence Enterprise Edition.

- Retail Insights provides packaged integration with select Oracle Retail source systems through Oracle Data Integrator (ODI) modules for extract, load, transform and aggregation processing.

- Retail Insights provides a flexible aggregation framework that can be easily customized after implementation to achieve an aggregation strategy that satisfies batch and reporting performance requirements.

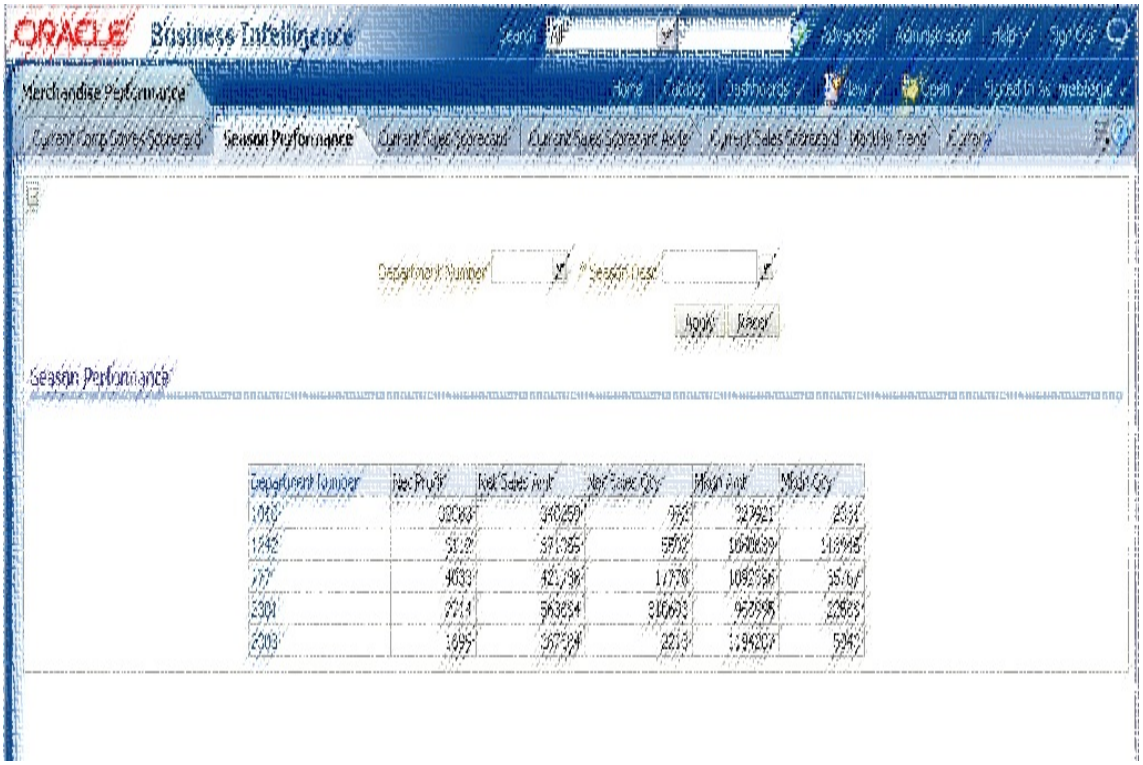

Figure 3.2: Retail Insights Testing Environment

- Dashboards and rich, high-performance metadata.
- Supports data for 18 different languages. The user interface is also translated into 18 languages.
- Retail Insights is packaged with sample user roles that leverage the Oracle BI EE security framework to manage dashboard and report access.
- Retail Insights ETL and reporting logic can be customized and extended by Oracle Retail Consultants to fit retailer-specific needs.[3]

Fig. 3.2 can give a rough idea about how Retail Insights Environment looks like.

3.3 Secure Global Desktop

Oracle SGD is a Secure way to access remote machine or a cloud-hosted application for an enterprise. It supports linux, windows, Solaris, Mainframe servers and certain Apple and Android Tablets as well. SGD is certified to be used with Oracle Products. This is essential, because if the database connection or its settings cannot be stored locally. Thus, people use SGDs to access a Remote Server and Setup different connections. I

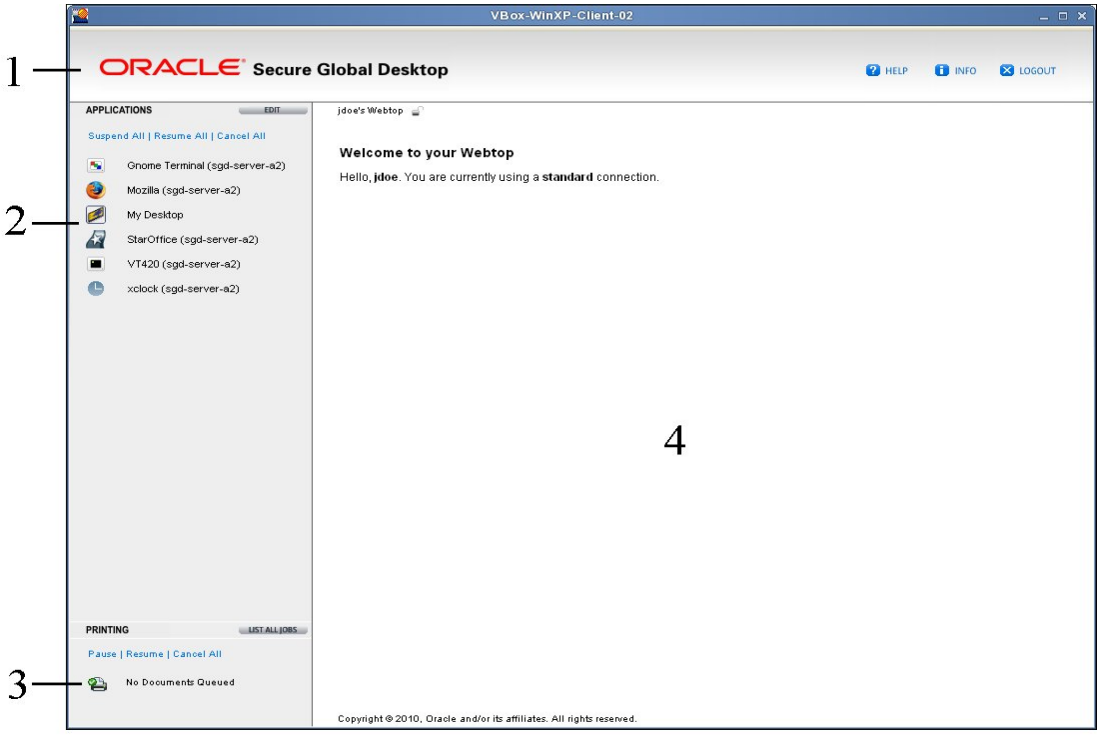

Figure 3.3: Oracle Secure Global Webtop

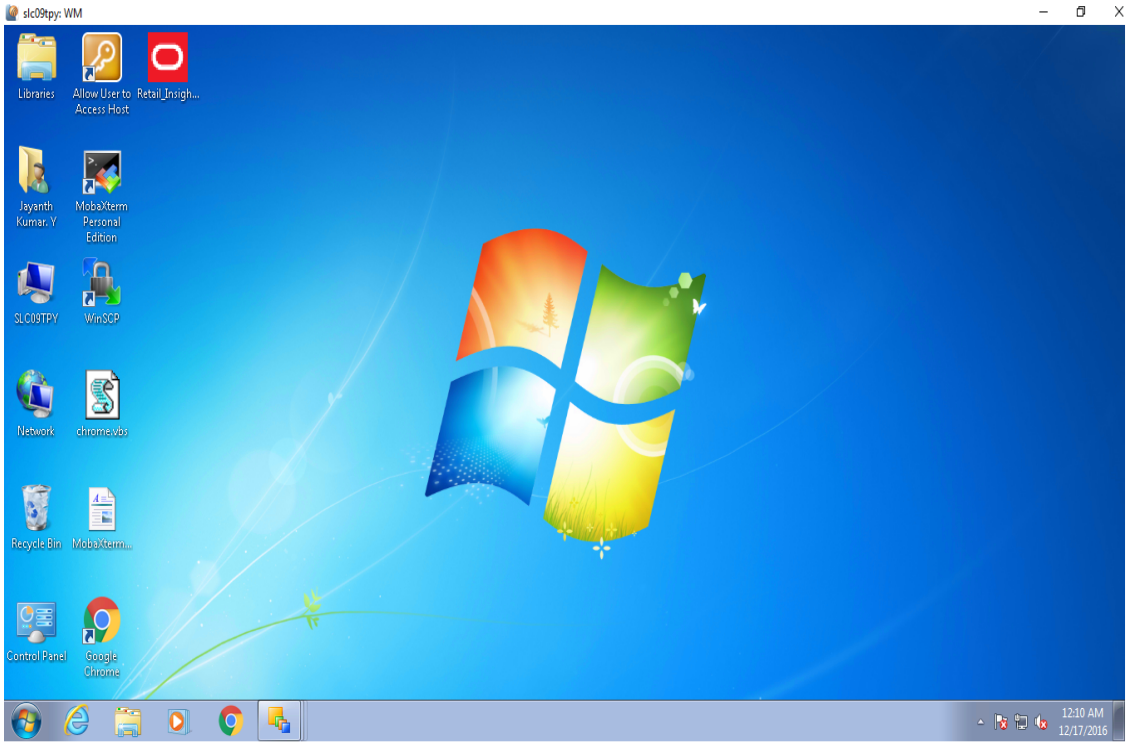

Figure 3.4: Oracle Secure Global Desktop

mainly use the SGD to access ODI Framework, as the Mappings for aggregation and filtering from base tables to Main Facts is stored over a Remote Server. This makes the connection faster, thus increasing our work speed. Fig. 3.3 and Fig. 3.4 show how Secure Global Webtop and SGD look like.

Description of Fig. 3.3 is mapped in Table 3.1 [4]

	Number	Description
1	Menu bar	Main Bar, including Help and Logout Options
2	Applications Bar	List of Applications that can be run by you
3	Printing Area	Enables you to control print jobs
4	Information Area	Information about the System and Error Messages are shown here

Table 3.1: Description of SGD Environment

3.4 Oracle Data Integrator

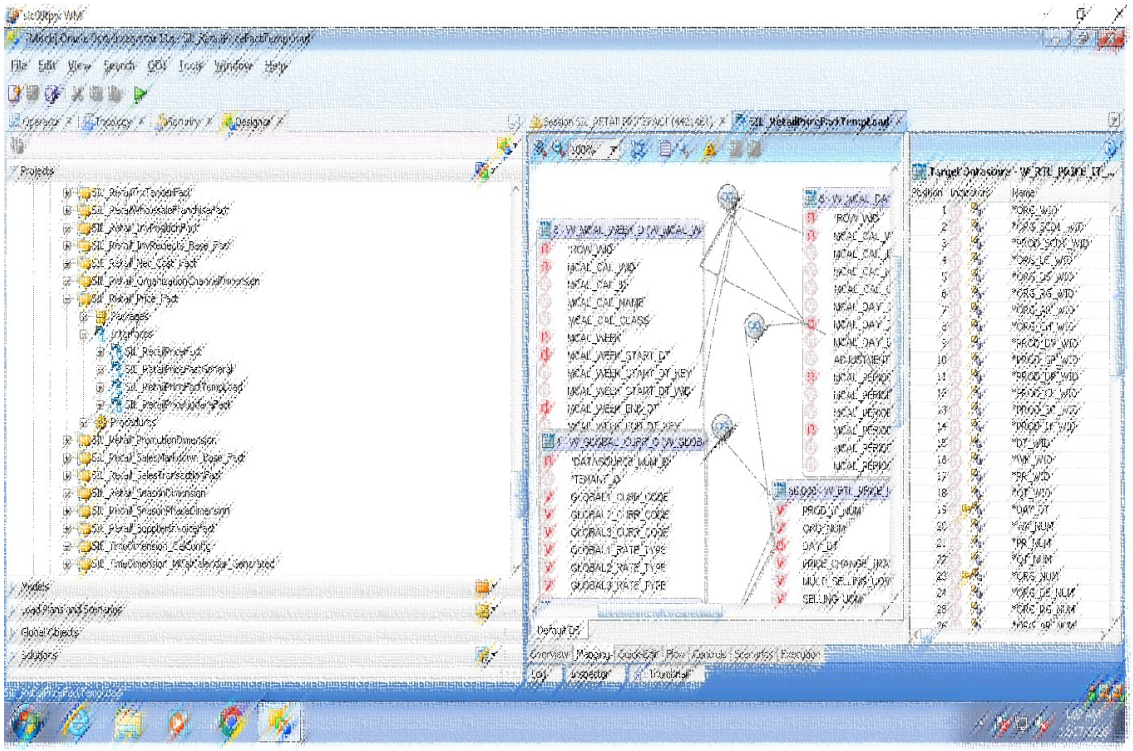

Figure 3.5: ODI Mappings

ODI is a wide-ranging data integrator which conceals within itself, the complex data integrations from multiple source tables, applies joins, filters, etc. and loads data into Source tables. In Fig. 3.6, you can see the ODI environment looks. ODI provides higher

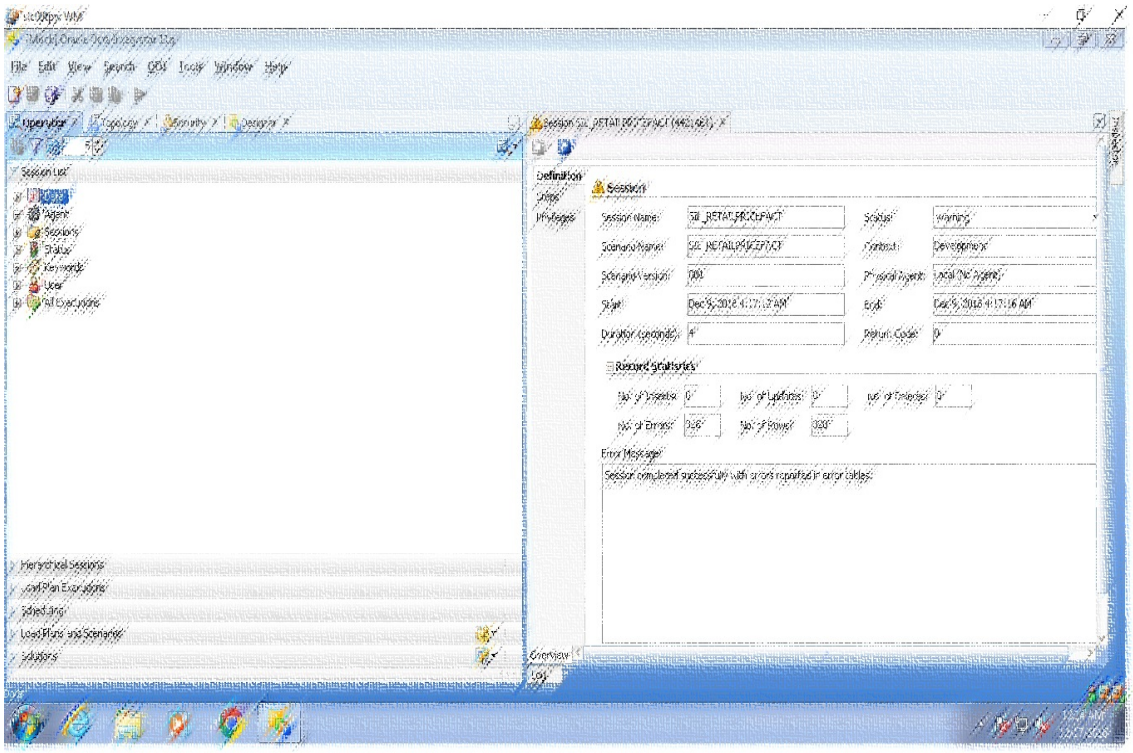

Figure 3.6: ODI Sessions

productivity and more secure connections with logs for each performed activities to monitor. It also allows users easier Interface to view the connections and mappings from and to tables being loaded. Fig. 5.2 shows Session history of a batch run.[5]

3.5 PuTTY

PuTTY is used to connect to multi-user systems using any of the SSH, Telnet or Rlogin methods. In Fig. 3.7, we have to supply Host Name, which afterwards connects to ask for your username and password and check for your privileges to log into the multi-user system. PuTTY uses Authenticode certificate for protecting alongside tampering during download. PuTTY consists of multiple modules, like PuTTYgen etc. But we only use PuTTY for our necessities like running scripts for loading data in different modules, by connecting to different servers.

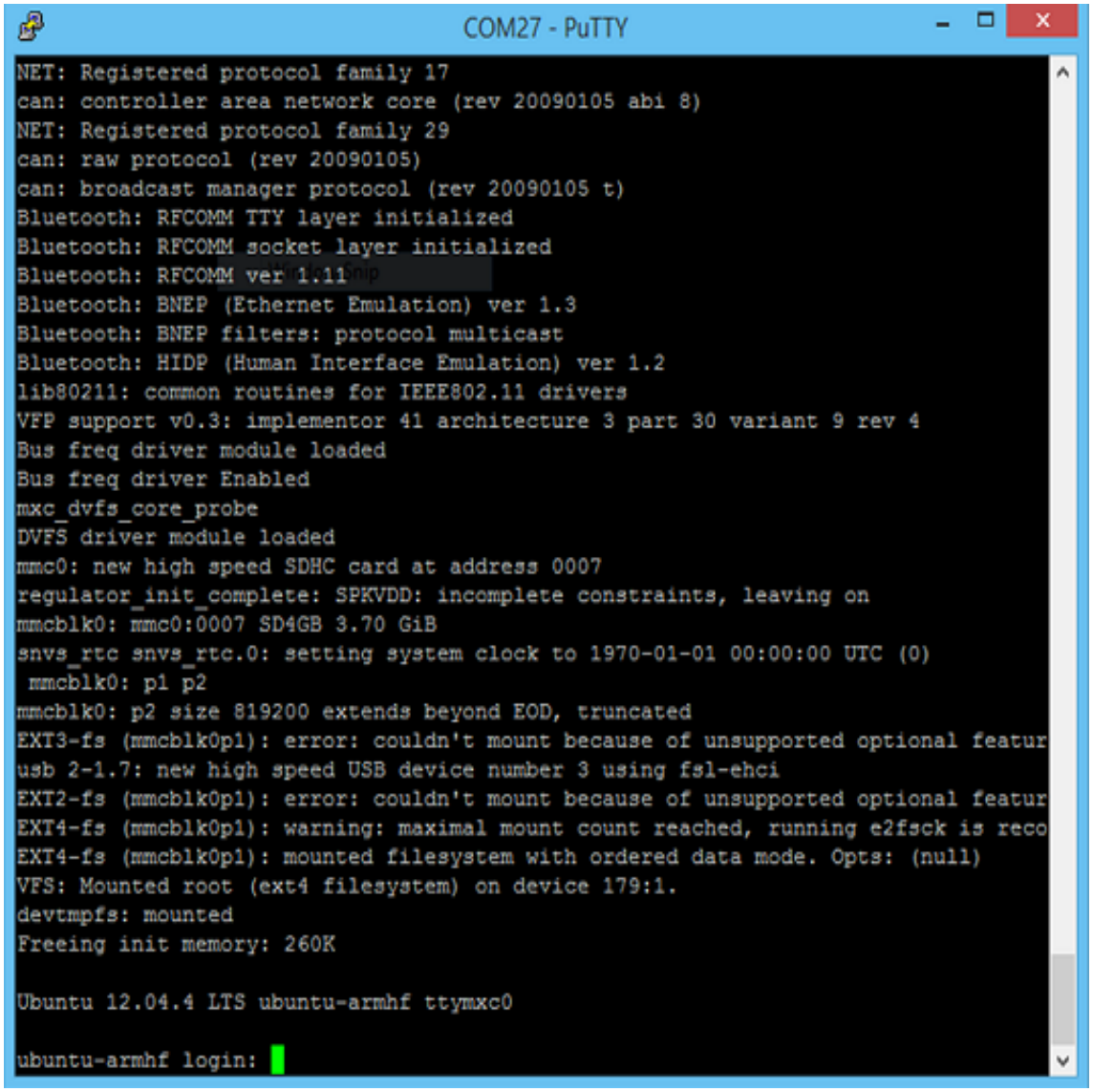

Figure 3.7: PuTTY Environment

3.6 WinSCP

Windows Secure Copy is used to securely transfer files between systems i.e. remote server and local machine. Standard FTP Protocols send your files and login information over the network as plain text, while WinSCP combine FTP with SSH tosend the information in an encryptred form. On the Left in Fig. 3.9 is your Local System, and on the right you can see the Remote Machine File Structure.

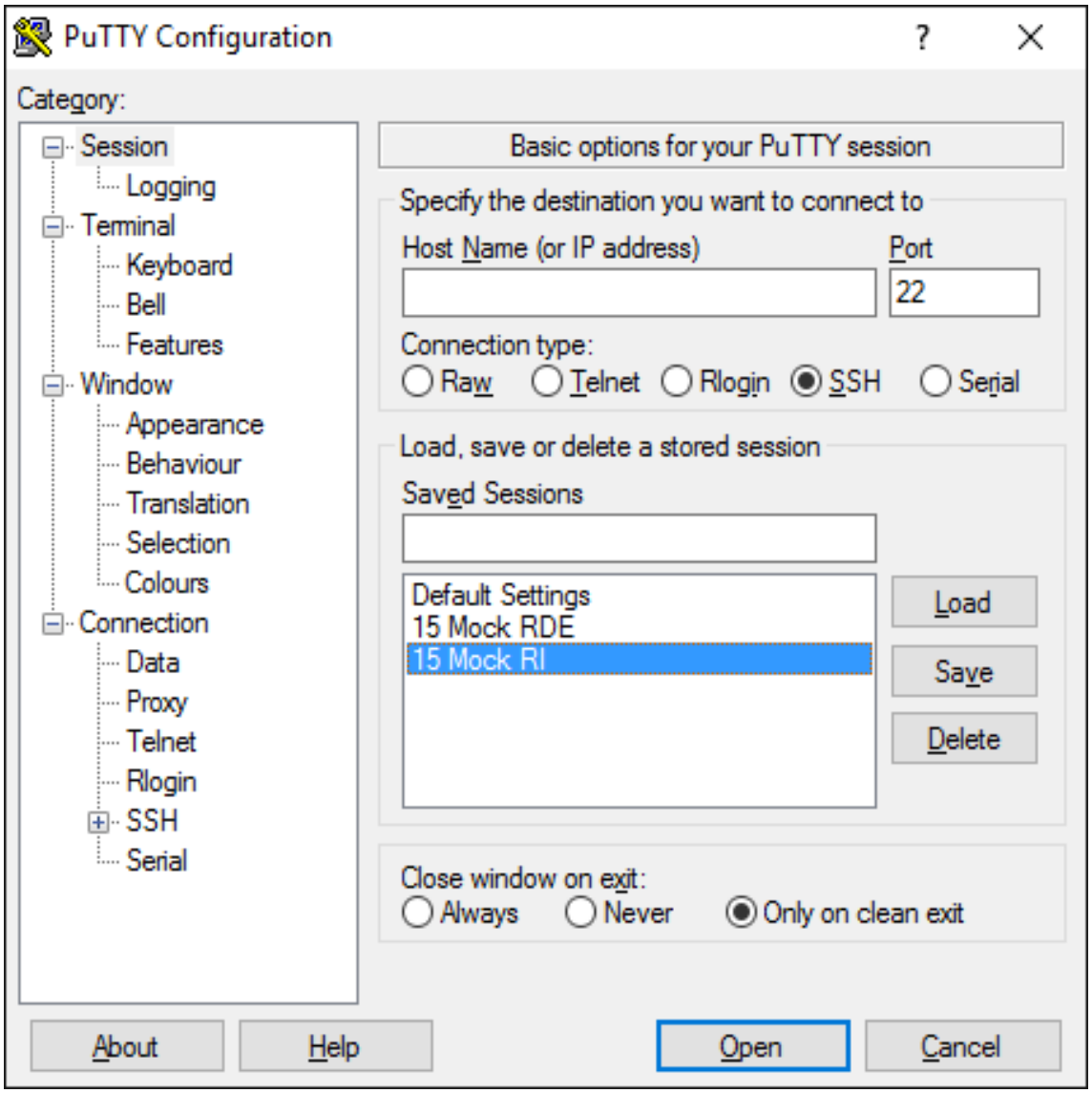

Figure 3.8: PuTTY Configurations Screen

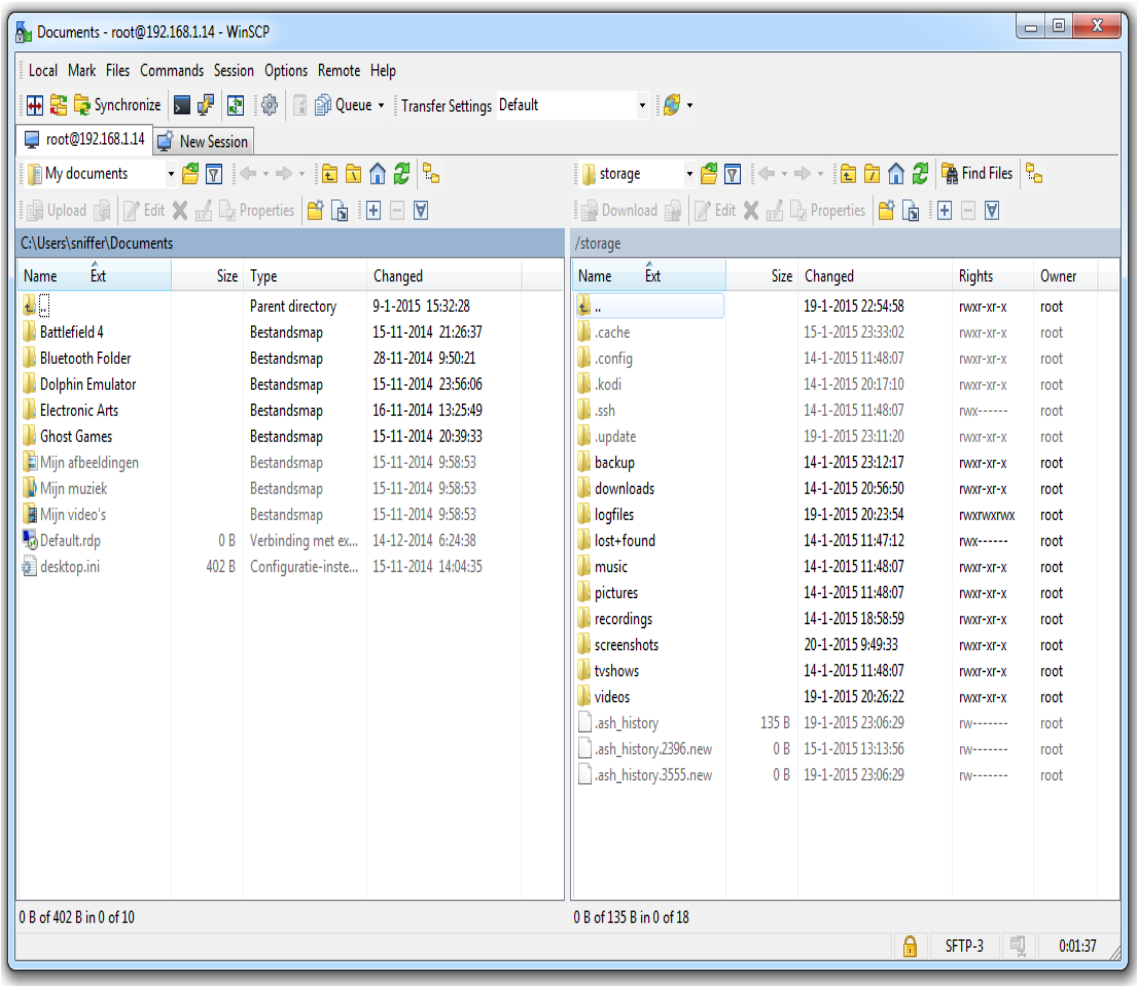

Figure 3.9: WinSCP Environment

Chapter 4

Technical Details

4.1 Tool Details

Tool	Version
OpenScript	12.5.0.3.960
WinSCP	5.7.7
PuTTY	0.67
Oracle SQL Developer	4.0.0.13
Secure Global Desktop	5.2

Table 4.1: Development Tool Versions

4.2 Minimum Requirements versus Actual Server Configuration

	Requirement	Configuration
Processor	2.6 Ghz	2.6 Ghz
RAM	8 GB	15.3 GB
OS	Windows 7	Windows 7
System Type	64 bit OS	64-bit OS
HDD	320 GB	500 GB

Table 4.2: Comparison of Server Configurations with Requirements

4.3 Minimum Requirements versus Actual Local Machine Configuration

	Requirement	**Configuration**
Processor	2.5 Ghz	2.5 Ghz
RAM	8 GB	16 GB
OS	Windows 7	Windows 10
System Type	64 bit OS	64-bit OS
HDD	320 GB	500 GB

Table 4.3: Comparison of Local Machine Configurations with Requirements

Chapter 5

Implementation

The work I did here at Oracle Retail as an intern for this semester includes Testing the loading of data in tables, resolving bugs in some products, and mainly automating the Report Generation and Testing of OBIEE and Automatic Data Generation for Tables.

5.1 Load Testing

Fact Tables are usually loaded from Transaction Tables. A warehouse consists of Facts and Dimensions for each particular portion. Transaction Tables are in a different Database, but within the same Warehouse. The reason for separating these tables is that these two are a part of two different systems, RMS and RI. RMS deals with transactional Data, each bit of it whereas RI needs more of specific data i.e., different records for suppose sales data, separate for Retail related records, etc.

So, Transaction Tables in one Database is used to load Fact Tables in RI. These loading portions cannot be done directly, so we have some Views which hold data joining transaction tables, and then we load them into Staging Tables. Staging Tables are just some database tables containing your company data in some or the other form. It is used to prep the data before loading into actual Facts.

Herein, questions might arise as to how are these two products any different than each other if they have the same data. The difference is that RMS is a transactional system (OLTP) whereas RI is an analytical system (OLAP). Retail Insights provides analytical reports to company to state the current state of their business, and accordingly, to take

necessary steps to increase their business.

5.1.1 RMS to RDE load

RMS has transactional and dimensional data within its different tables. But just a history of transactions is not enough for a fast paced business. Companies need to see this data in a more subtle way. We cannot directly link RMS to RI with a database link because of the following reasons:

- Oracle RI was previously an on-prem application, but then it moved to cloud service. RMS and RI being on two different sources, we cannot directly connect them.
- The data format used in both of these products is different.
- To ensure security of both the products.

So, what oracle did was, they created a middle-ware product called as the Retail Data Extractor. RDE consists of SDE Mappings (Source Dependent Extracts) which reads data from RMS and creates flat files which could be used in RI for loading the Staging Tables. These SDE mappings have joins between multiple RMS tables, and mappings to load data correctly.

Staging Tables load data into Facts using some Temporary Tables, to reduce the number of joins at a time. Certain Joins are done while loading Temp Tables, and other during Fact Loads. But, what happens when a load fails? Those data are loaded into Error Tables for debugging, and figuring out what possibly went wrong.

As can be seen in Fig. 5.1, RMS data is loaded using an Extract, Transform, Load Process from SDE Mappings into RDE Staging Tables, which in turn is used to create staging flat files using Universal Adapter SDEs to load into RI tables.[6]

5.1.2 RDE to RI load

Once we get data in RDE and generate flat files, those files have to be transferred to RI environment and then load it into staging tables using universal adapters. The SIL programs then load these staging table data into Warehouse Tables. This flow can be seen in Fig. 5.2.[7]

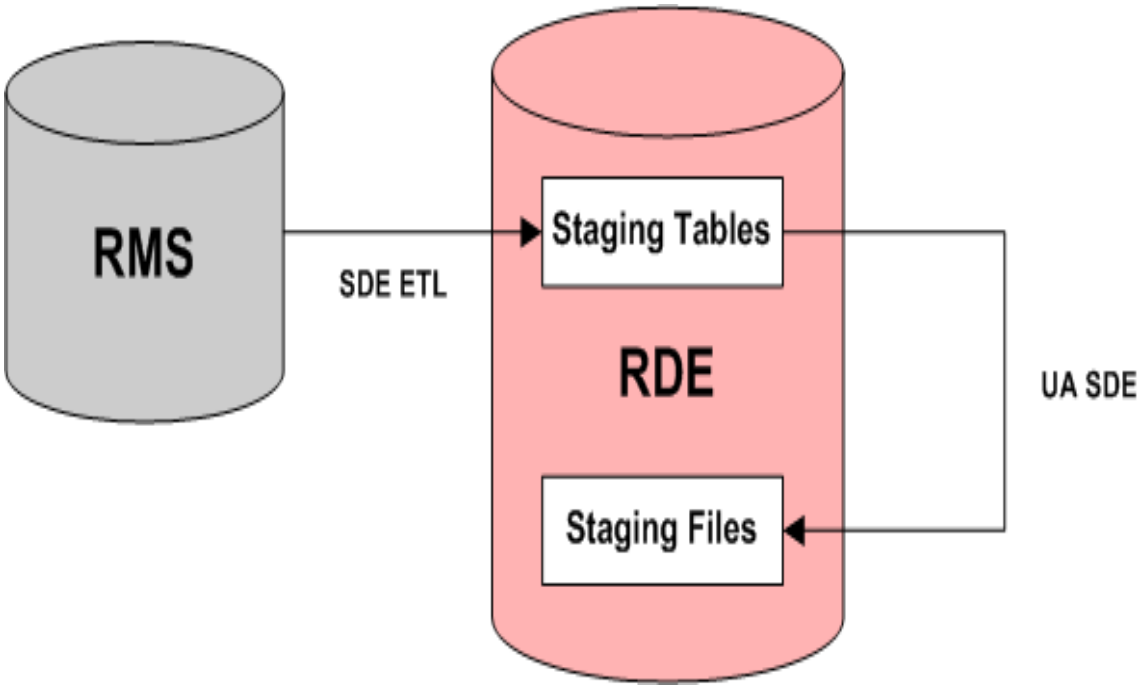

Figure 5.1: RMS to RDE Data Flow

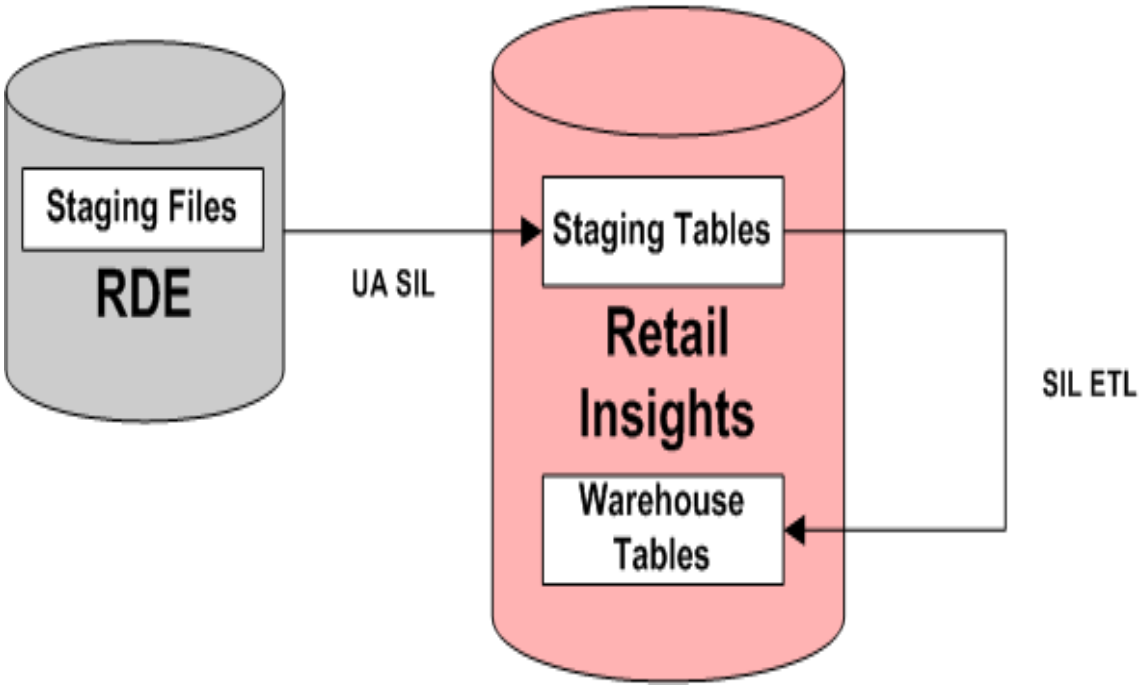

Figure 5.2: RDE to RI Staging Data Flow

5.2 Report Generation

Once the RI tables are loaded, we have to test the metrics that are defined in OBIEE. Opening the OBIEE analytics will ask for a login as in Fig. 5.3.

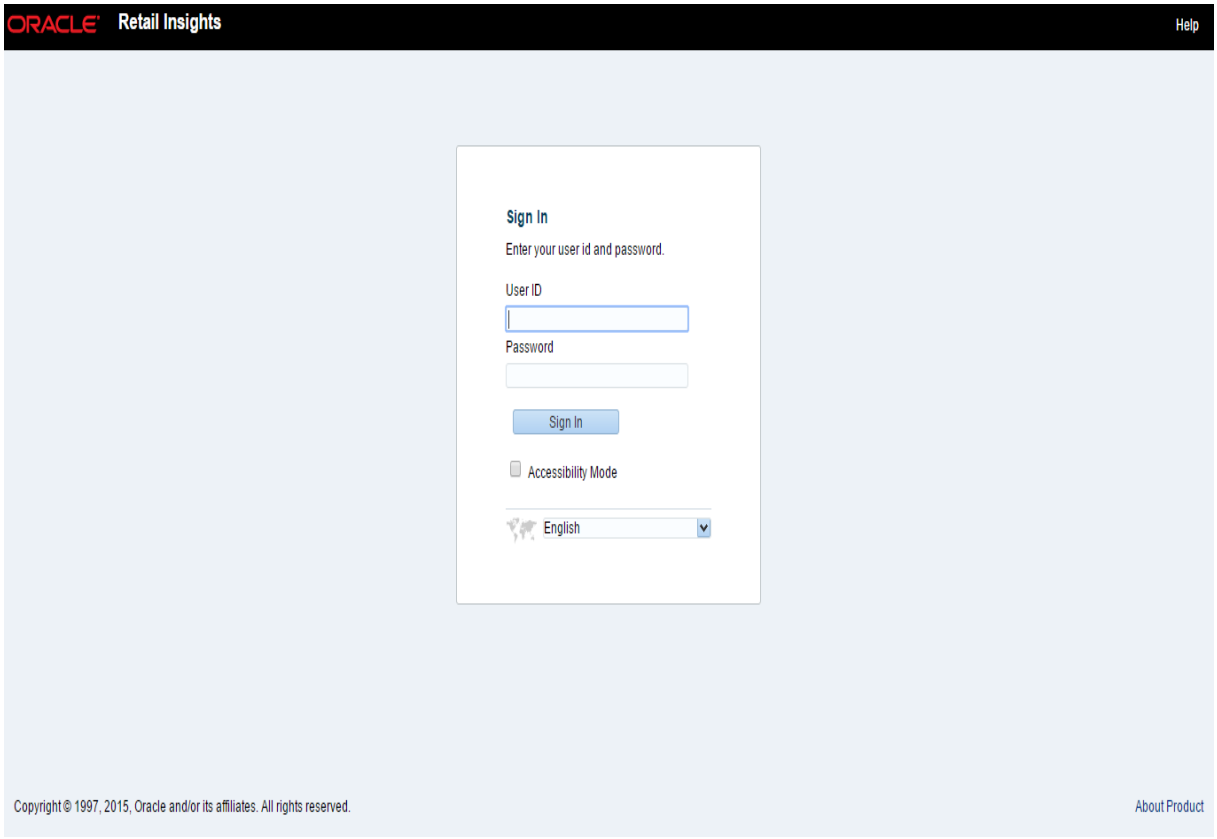

Figure 5.3: OBIEE Login

As logged in, you'll have to select New -¿ Analysis (Fig. 5.4) and then the subject area as per the scenario as in Fig. 5.5

This entire flow will then redirect you towards creating a custom report by pulling those metrics in the selected columns part. For a sample, I've pulled in a metric with some level attributes in Fig. 5.6.

Fig. 5.7 shows the results that are pulled from the database tables that we load into RI Fact tables. This process is automated and verified using the program that I created, and is mentioned in detail in section 5.3.

5.3 Automation of Testing Process

I Mainly created following scripts which provides a base for Automated Integration in Retail Insights:

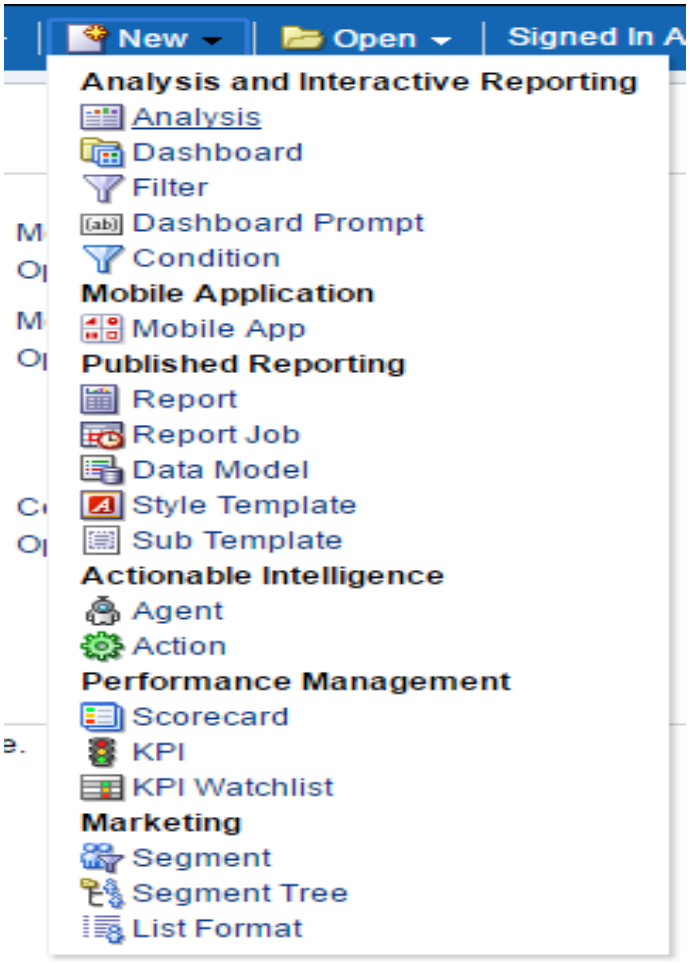

Figure 5.4: New Analysis

5.3.1 IntegratedAutomation

The Testing starts running through this main Script which describes whether data generation, report generation, etc are needed or not, and does the work accordingly.

5.3.2 FileAccessLibrary

FileAccessLibrary acces and stores files which have metrics along with its formulas and table name which they will hit. This information makes report generation easier.

5.3.3 DbAccessLibrary

The main purpose of this script was, as it's name suggests, to handle all Database related Tasks, from establishing DB connection, to Accessing and updating databases, to closing those connections.

Figure 5.5: Subject Areas

5.3.4 DataGeneration

DataGeneration Script generates data using the shell scripts defined in Linux Servers by users. It also makes use of FileAccessLibrary to determine which table needs data generation. The data generation erases previous data before generating new one. This file is yet under development, and is not complete

In Fig. 5.8 is shown a graph, which is roughly generated using Reports created using some metrics.

5.4 Resolving Bugs

There could be minor bug in a product at some time. A bug does not necessarily mean error, it could be an incorrect result caused due to some fault, or some feature not implemented at all. When such a bug is encountered, a new entry for that bug is made. The

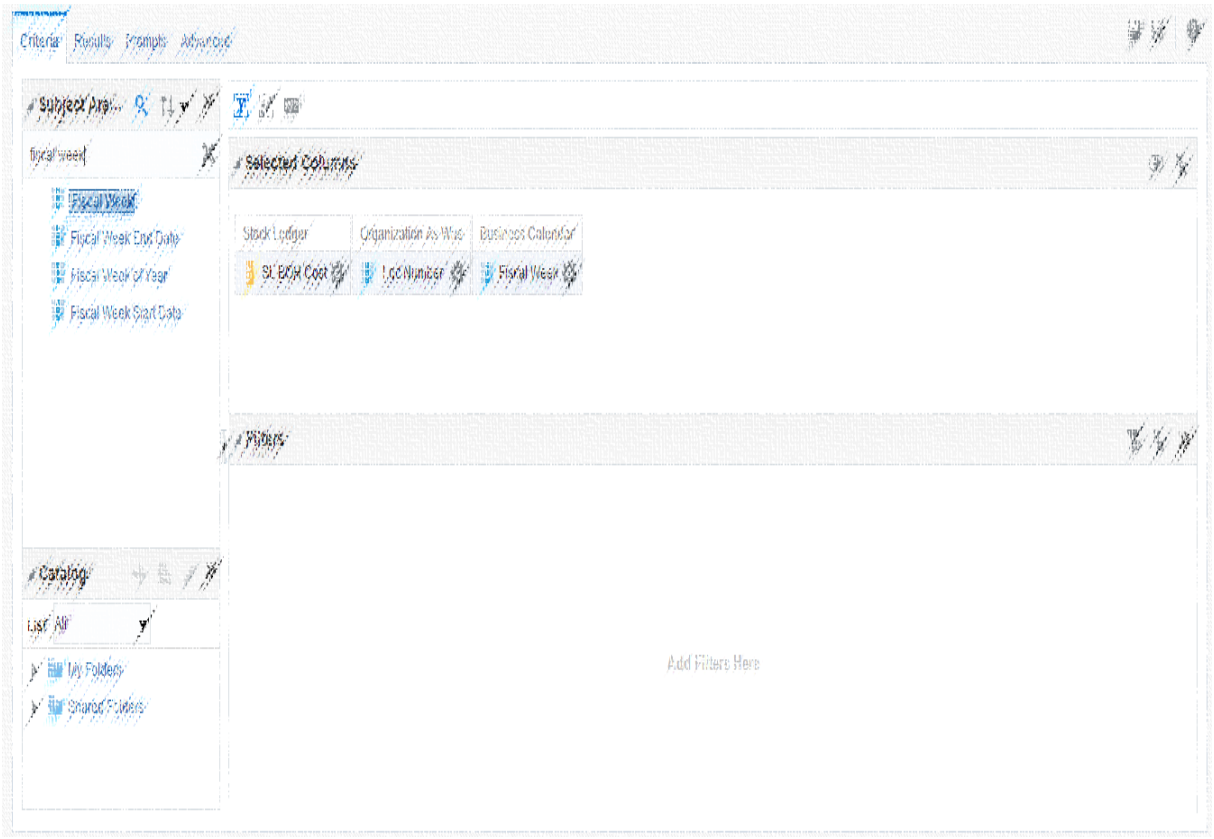

Figure 5.6: OBIEE Criteria for a Subject Area

first step is to recognise the exact LOC where that error is occurring. Now, from these many joins and tables, figuring out that is pretty tough, as you need to follow each step manually, that is done through scripts before. And lookout for errors in each step.

We have different batches for loading Transactional Data into Staging Tables, and an entirely different set of programs for loading the data from staging to facts. So the first step to check is whether staging table is loaded with the data we want. If that is correct, the second set of batch programs that load staging data to fact, has some bug. Otherwise, the programs loading transactional data to staging table are having a bug.

These bugs could be classified in a varied manner. A brief classification can be seen in Fig. 5.10.

After figuring out the set of programs to look at, we have to drill down to the one program that loads data for the particular table wherein incorrect data is stored. Sometimes, the bug could be due to multiple mappings, and sometimes even one mapping, or

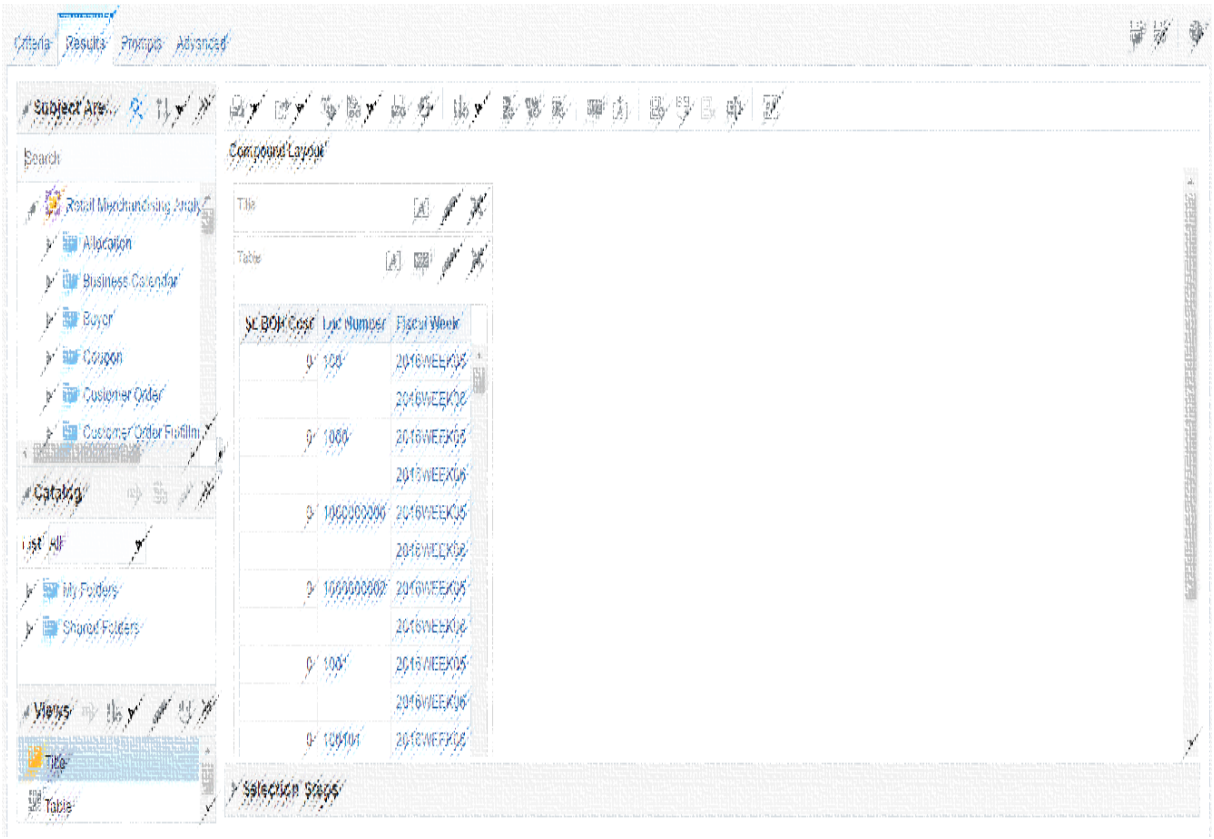

Figure 5.7: OBIEE Results for the selected Metrics

even some join condition or Filter Conditions.

These mappings and Programs are all stored in ODI and is accessed using SGD. Using PuTTY, we access and run Scripts to load data from Staging to Facts Manually.

5.5 Automating BugDB to Confluence transfer

When these Metrics are tested, we do Data Validation, and Formula Verification. The entire flow is deplicted in Fig. 5.10. If for any test case, these metrics fail, we log a bug to notify about that to the developers so that they can correct the mistake and reload the metric. This data was all done in Oracle's BugDB service. But as we progressed, there came in a new product called as confluence. So at first all the bugs were manually entered into confluence page for summary. So we wrote a script which could connect to BugDB, fetch out the related bugs, and automatically create a confluence page for the bug summary. This helps in saving a lot of time that is wasted in manual updation on the confluence page.

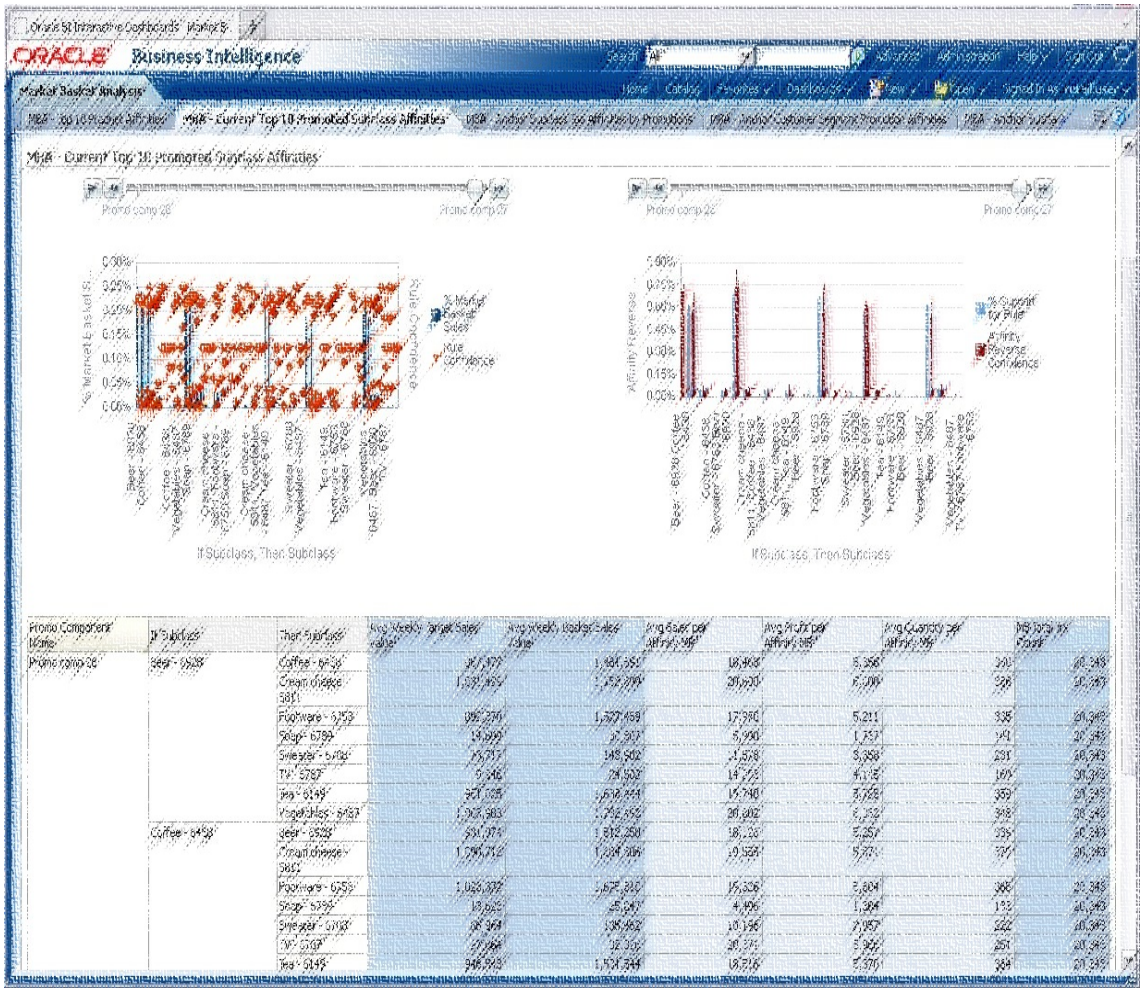

Figure 5.8: Analysis Graph in RI

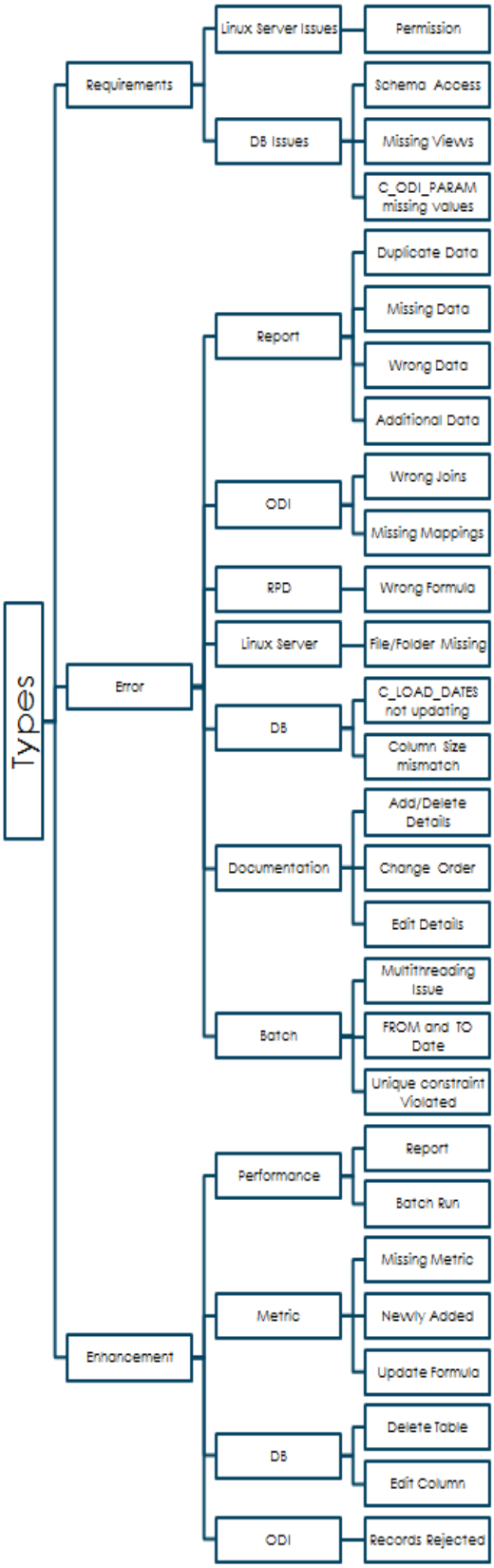

Figure 5.9: Error Classification

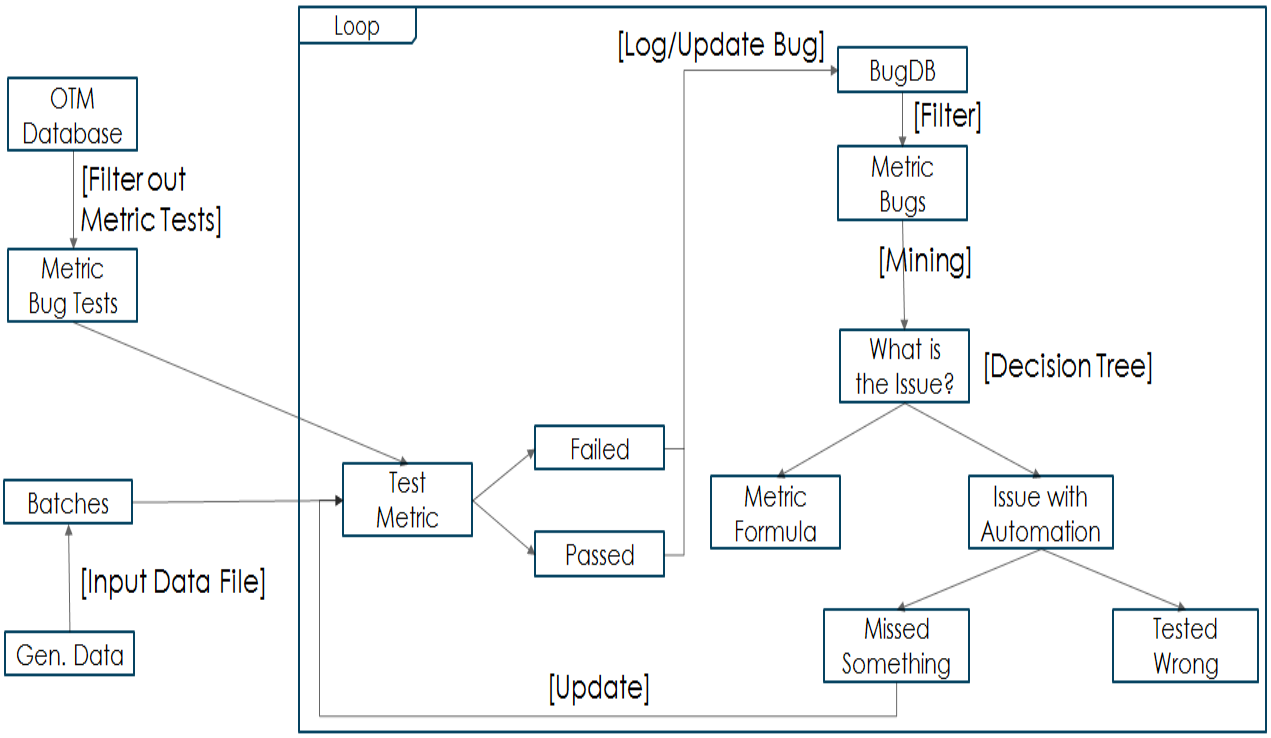

Figure 5.10: Metric Test Flow

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

After joining Oracle Retail, I have undergone Trainings for OpenScript, and have studied documentations for the same. I was assigned task to test the metrics defined for a company and automate this entire process which was completed successfully. Also, I have undergone some other basic company needed trainings about the Security for their products and Oracle Application Testing Suite.

I have also generated certain reports for those metrics which I cannot show due to the Oracle Retail Confidentiality Rule. But I have developed and improved the basic Report Generation and Report Testing Automation Tool which has helped us in testing the metrics faster than before. I also automated the Data Generation Process in these Staging tables. Other than this, I've worked on Quality Assurance of the product to make sure the system is reliable ans secure.

6.2 Future Scope

Optimization is something that can be done in every system, even if it is already a bit optimized. We need to optimize the automation process of report testing and generation also, as far as possible. Also due to this project, I can say this, that the process where metrics are created could also be automated by some modifications to the code. That would mean just manually making some Excel Entries and running the modified automation program to create those metrics.

Bibliography

[1] R. Santos, *OpenScript User's Guide*. Oracle, May 2012.

[2] *Oracle Retail Insights Predictive Application Server Release Notes*. Oracle, December 2015.

[3] "Oracle Secure Global Desktop." http://www.oracle.com/technetwork/server-storage/securedesktop/overview/index.html.

[4] "Oracle Secure Global Desktop Documentation." https://docs.oracle.com/cd/E19351-01/821-1925/z40000d11283446.html.

[5] "Oracle Data Integrator." http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html.

[6] "Oracle Retail Data Extractor Documentation." http://docs.oracle.com/cd/E82085_01/rde/150/html/implementation_guide/rde-imp-universal-adapter.htm.

[7] "Oracle Retail Insights Documentation." https://docs.oracle.com/cd/E22577_01/pdf/150/ri/html/implementation_guide/ri-imp-universal-adapter.htm.