Chapter 4: SOA based Heterogeneous Data Migration Tool

4.1 Motivation behind HDMT

The Indian Education sector has seen a tremendous growth in the last decade with the Ministry of Human Resource & Development (MHRD) giving permission for private institutions / universities for higher education. This required the setting up of various statutory bodies like All India Council for Technical Education (AICTE), National Board of Accreditation (NBA), National Assessment and Accreditation Council (NAAC), Directorate of Technical Education (DTE) etc. These bodies and their policies enabled the establishment and formation of various institutes in almost every state which required their monitoring & control, to ensure uniformity of environment; both administrative as well as technical. Effectively, they provide the necessary rules, regulations and guidelines in the formation and running of any technical educational institution so as to be a world class organization in leading the technological and socioeconomic development of the country. This is done by enhancing the global competitiveness of technical manpower and by ensuring high quality technical education to all sections of the society. Various MIS / BI reports are published to gain more insight into the sector. Such reports are generated by gathering relevant data of various institutes or colleges. These reports show the growth and volume of courses and educational institutes.

This data is made available by respective colleges or institutes to the governing body website in their prescribed format for better control and monitoring of the environment. The data is mainly related to the institute or college infrastructure, financial investments, teaching and non teaching staff, student admitted, courses conducted etc. This data is obtained and published as various reports for the purpose of better human resource planning and development by the governing body.

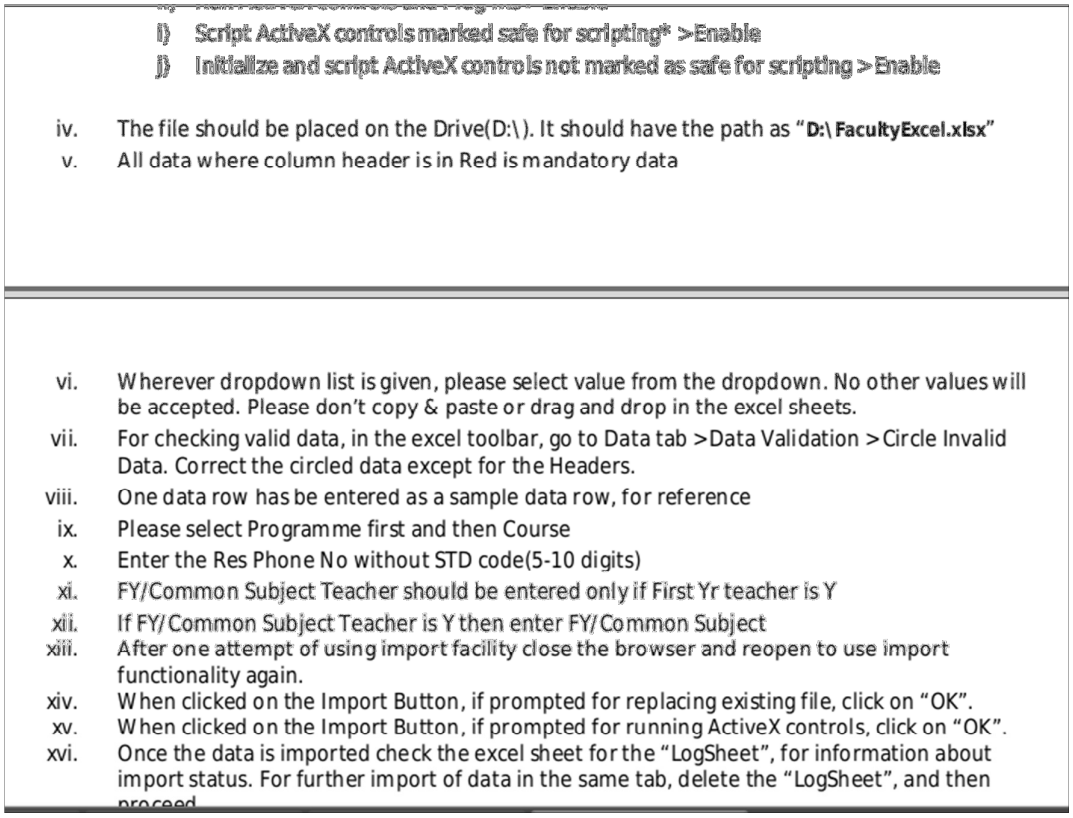
On the other hand, the colleges or institutes try to maintain the data of such large volume by implementing software systems or by other methods at the college level. Currently, few of the colleges or institutes have their individual MIS or ERP system implemented at their end so as to handle vast amount of data and cater to the demand of any management reports. The software systems are designed to be implemented as per the specifications of the colleges and need to provide for any future changes in the current implementation. Many colleges use their respective software systems or other software methods which means that their entities or data attributes are very much similar and the schema may vary in naming convention, data type and data size because of vendor and requirement differences. But, the information that they want to exchange with the governing bodies is definitely the same which needs to be converted into prescribed format before upload process at governing body website.

Currently, these two systems communicate or exchange their data with each other by using some middle entity like excel for the data upload. The excel sheet has to be as per the prescribed details as shown Figure 4.1. The column headers, certain standard and defined input like course have be selected from the options given. The sequence of columns and the format of the data is to be followed as prescribed.

Title	First Name	Middle Name	Surname / Family Name	Mother's Name	Father's Name	Res Phone
Miss	ANU	MUKUND	BILONIA	ARCHANA	MUKUND	24563214

Figure 4.1 Sample of format of excel sheet

The instructions mentioned for the upload process is also mentioned in the user manual provided by the governing body. A sample of the instructions is as given in the Figure 4.2, Figure 4.3 and Figure 4.4



i) Script ActiveX controls marked safe for scripting* >Enable
j) Initialize and script ActiveX controls not marked as safe for scripting >Enable

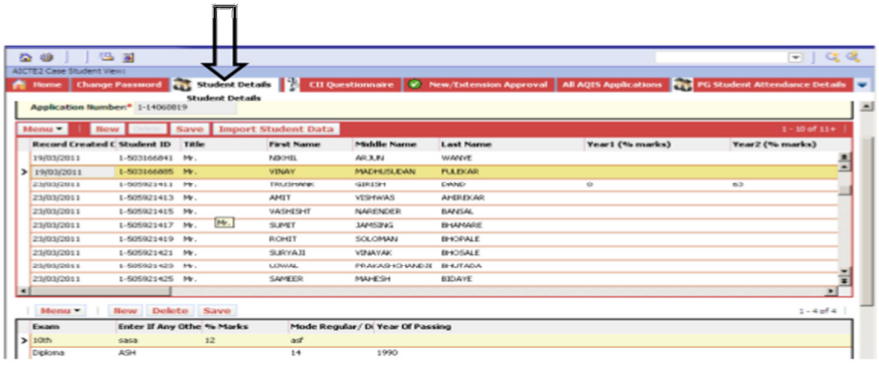
iv. The file should be placed on the Drive(D:\). It should have the path as "**D:\FacultyExcel.xlsx**"
v. All data where column header is in Red is mandatory data

vi. Wherever dropdown list is given, please select value from the dropdown. No other values will be accepted. Please don't copy & paste or drag and drop in the excel sheets.
vii. For checking valid data, in the excel toolbar, go to Data tab >Data Validation >Circle Invalid Data. Correct the circled data except for the Headers.
viii. One data row has be entered as a sample data row, for reference
ix. Please select Programme first and then Course
x. Enter the Res Phone No without STD code(5-10 digits)
xi. FY/Common Subject Teacher should be entered only if First Yr teacher is Y
xii. If FY/Common Subject Teacher is Y then enter FY/Common Subject
xiii. After one attempt of using import facility close the browser and reopen to use import functionality again.
xiv. When clicked on the Import Button, if prompted for replacing existing file, click on "OK".
xv. When clicked on the Import Button, if prompted for running ActiveX controls, click on "OK".
xvi. Once the data is imported check the excel sheet for the "LogSheet", for information about import status. For further import of data in the same tab, delete the "LogSheet", and then proceed

Figure 4.2 General Instructions for accepting and sending data using Excel sheet

Student Details:

1. The institute log's in to the portal.
2. Navigate to – **Student Details** View.
3. Click on the new button to create new student record.
4. To create new students for academic year 13-14 click on New button. To search for the students for academic year 12-13 you can search in the existing view for the student as the record has been already created previous year.

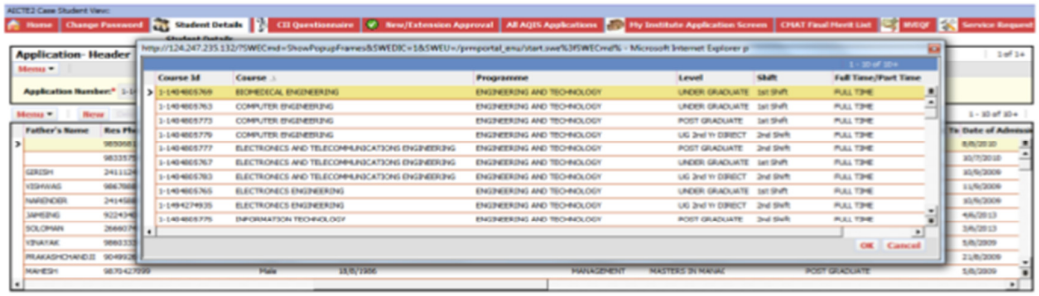


5. In Student Details view the fields that are mandatory are, First Name, Last Name, Title Father's Name, Mother's Name, Student Status, Date of Birth, Date of Joining, Course, Programme, Permanent Address Line 1, Home District of the student, Home State of the Student. Once student selects the course from pick applet Course Id, Programme, Level, Shift gets populated automatically.

Figure 4.3 Instructions for adding student details

Help Manual for PG Approval Process

6. The Institute can select the approved course from the pick applet and its corresponding programme, level, shift, full time/part time and course id get populated automatically.



7. Institutes can import student records in bulk by clicking on the button 'Import Student data'. You need to save the Student Excel Sheet in D:\StudentExcel.xls Update the StudentExcel.xls with all student records that need to be created and click on 'Import Student data button'. This will create records in Student Details View.

   Student Excel Sheet for Importing Student Details is available in AICTE Website -> Students ->Scholarships ->PG Scholarship GATE/GPAT ->Format for importing student Data

Figure 4.4 Instructions for importing student details from excel sheet

After the upload process, there is a report on the number of records sent and updated. This report helps in identifying the success of the upload operation. Any errors or exceptions are mentioned in the report and have to be accordingly handled by the colleges or institutes.

The entire process of data gathering, data formatting and data uploading or sending, generated an interest in me. My work on one of the module for sending data from college to governing body on their website motivated me to think about the situation and the complexities therein. The most important question that came to me was that if these two systems are having exactly the same data then there should be no need of such middle ware like excel. Data should be transferred by some easy methods.

Alternately, in cases where colleges are using the software systems, then, for the data uploading, the college has to download the data, convert in required format from the software system that they are using and then upload it on the governing body location. The trouble with this middle ware is the need to download the data for its exchange purpose; which spoils the security, authorization, privacy etc. that has been implemented on the MIS system to maintain above data. This sequence of activity is an additional burden to the DBA and ideally should not require an intermediate data source for the data communication.

All these facts along with the discussion in 3.2.2 (d) related to the communication helped me in identifying the features of a tool for the above situation. My study about various DBA tools as discussed in chapter 3, gave an idea about the existing migration tools and its features and also about the existing ETL tools. These tools are complete in themselves with their respective objectives and features. As of now, the entire DB migration is already being done successfully by many DB tools like SwisSQL Data Migration Tool as discussed in 3.3.4.1. So, I have focused on the requirements of the data migration / exchange tool with respect to education sector or any other sector in which data is to be migrated between distinct companies / organizations with some mutual agreement to share details related to the data exchange. **The tool is a variation of existing migration tools where DBA can actually select the attributes and records and not carry out the entire DB or table migration.** I have also taken into consideration the fact that different companies / organizations maintain data using different technology. This means that the proposed tool should implement data exchange / migration between heterogeneous DBs.

The understanding about SOA and Cloud services, together; also pointed to the fact that there could be a service made available online for the data transfer or data uploading in such cases. The service can greatly benefit all those involved in the data exchange activity. Viewing this requirement as additional services to be made available to DBA of institutes or person handling DB activities of institutes or institutes' software, an additional layer of

service was planned as it involves large amount of data transfer between the affiliating institutes and governing bodies. **This transfer necessitates in most of the cases, the format and data type transfers which becomes a bottleneck for DBA. The tool is handy in such situations.** Keeping all this in view, *I have implemented a SOA based Heterogeneous Data Migration Tool (HDMT), which will aid the DBA in the task of exchanging or migrating the required data.* The HDMT tool aims to provide a general utility for the easy exchange of data among the distinct organizations or governing bodies who have similar data and need to exchange. The tool has been developed with SOA approach in mind. The large amount of data may involve a distribution of data as in DDB. The data in DDB could be kept vertically or horizontally fragmented. All these are the requirements for a DDB administration tool for DDB which have been implemented in HDMT. The HDMT tool proves the heterogeneous data migration process for DDB and can be tested and implemented further for practical benefits.

4.2 HDMT Purpose

The main purpose of HDMT is to demonstrate the migration of data between distinct heterogeneous DDB servers where the data can be selected as to the most detailed column and record level for migration. This tool facilitates the DBA to migrate the data from a source to more than one DB name / Table name / Column names / Compatible types without writing any configuration code. It implements the data migration in terms of horizontal and vertical fragmentation which is an important feature of DDBs. This facility is different from the existing DBA tools used for data migration by the DBAs and no middle entity like excel is required. It is also different from existing ETL tools as it requires data transformations with respect to the data type only. Also, the fact that ETL tools are existing for the purpose of data warehouse and mostly are scheduled to run at given time, which is different than the HDMT. In order to migrate the data in case of DDB, the HDMT requires few inputs which have to be given to get the desired results of data migration. The HDMT user interface is as shown in Figure 4.5 and Figure 4.6.

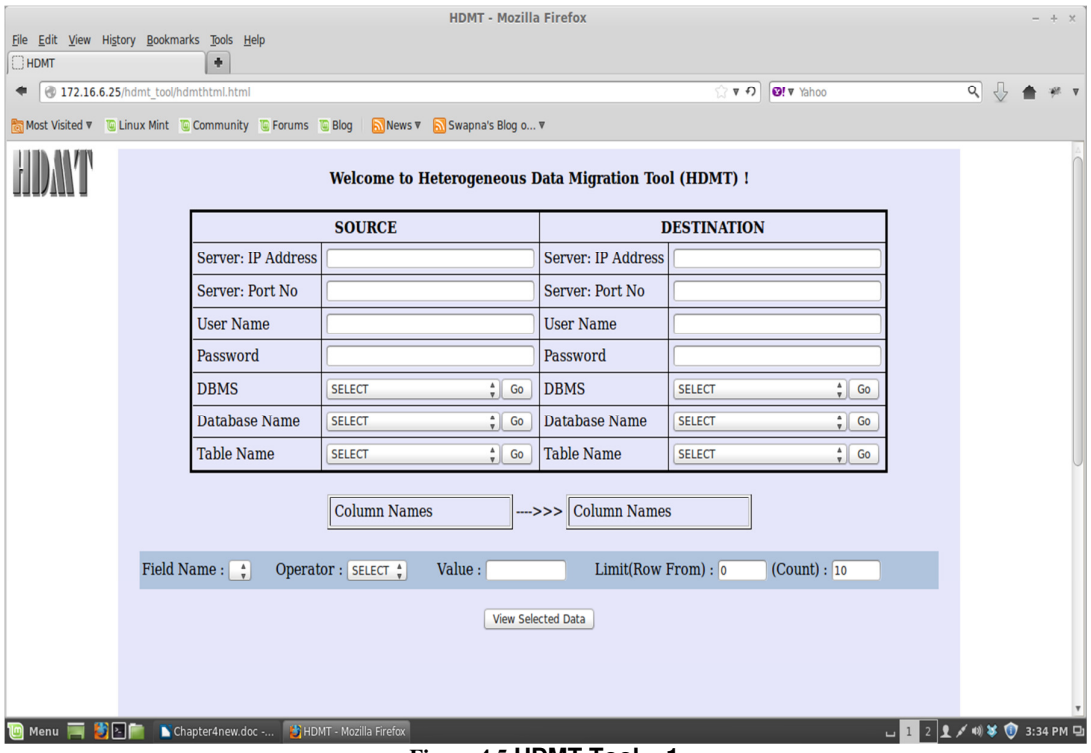


Figure 4.5 HDMT Tool – 1

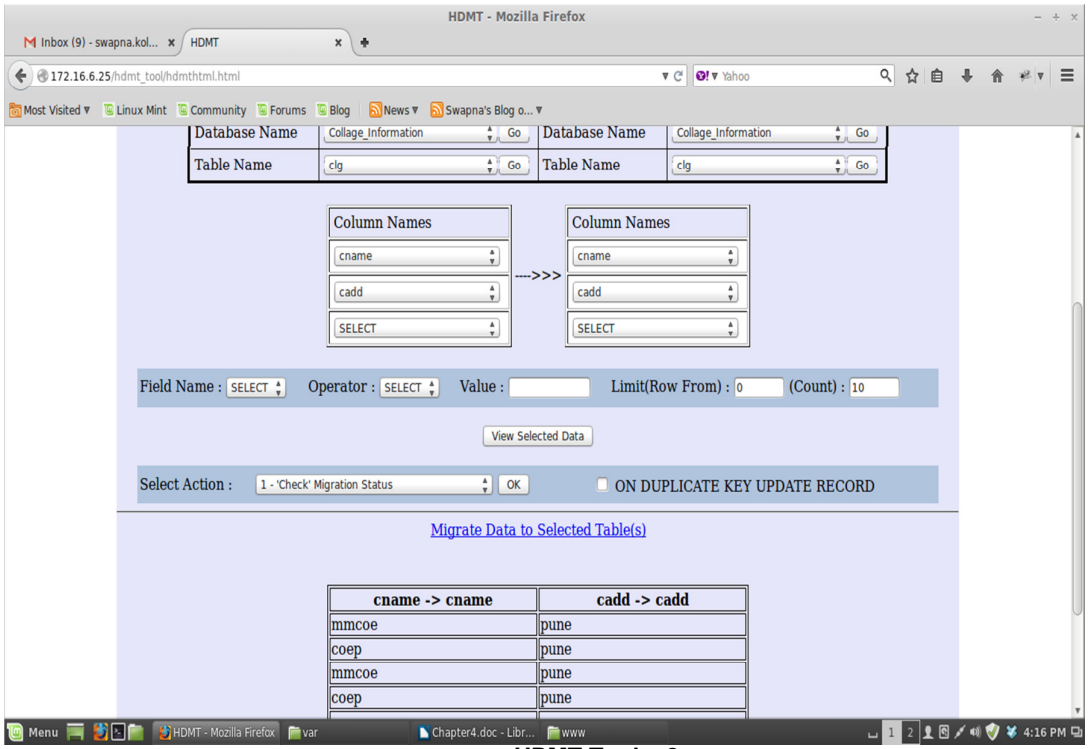


Figure 4.6 HDMT Tool – 2

4.3 Mathematical Model for HDMT

The HDMT is developed for the purpose of demonstrating heterogeneous data transfer in a DDB environment. It is based on the core functionality of selecting the desired data from source (so as to define our focus for horizontal fragmentation), transforming the selected data according to compatible data types (this is needed due to the data migration between heterogeneous databases) and finally migrating it to selected destination table(s) (in case of vertical fragmentation done in DDB). Accordingly, the mathematical model based on relational algebra expression is shown for the three functionality or processes as given below.

4.3.1 Selection of source data

Let us suppose that,

T = *table name*

D = *database name*

S = *database server name*

c = *condition*

CL = *column list or attribute list*

S is of a given RDBMS such as MySQL / PostgreSQL. S has many databases denoted by D. A database has many tables denoted by T. Hence, $T \subset D \subset S$ is true.

Let A be the sum of all data D_i (*0 to n; where n is the number of records*) belonging to the set of selected attributes (denoted by CL) and tuples (denoted by the condition c) of T which is a subset of D which is a subset of S.

$$A = \sum_{i=0}^{n} D_i \in (\prod_{CL} \sigma_C(T \subset D \subset S)) \quad \text{.....} \quad \textbf{Equation 4.1}$$

Let B be the sum of all data D_j (*1 to m; where m is the number of records*) belonging to the set of selected table name, T of database D which is a subset of server S.

$$B = \sum_{j=1}^{m} D_j \in \prod_{T} (D \subset S) \quad \text{.....} \quad \textbf{Equation 4.2}$$

Let C be the sum of all data (*1 to l; where l is the number of records*) belonging to the set of selected database D of server S.

$$C = \sum_{k=1}^{l} D_k \in \prod_{D}(S) \quad \text{..... Equation 4. 3}$$

Then for the entire data that can be accessed through A, B and C, the following is true.

$$A \subset B \subset C \quad \text{..... Equation 4. 4}$$

4.3.2 Transformation

This process requires transforming the selected data according to compatible data types so as to evaluate whether such compatibility on data type is possible or not. If it is possible then user can proceed to next step discussed in 4.3.3 else user cannot proceed to next step mentioned in 4.3.3 with appropriate message and the possible reason of incompatible data type conversion. In case of compatible data type conversion it is onto the user to check whether to proceed with the migration. So user has to check that unnecessary truncation of data does not take place even if data type is compatible. This is to say that small int stored as big int is possible but vice versa may not be appropriate due to data truncation. Hence, compatibility has been given for similar data types as shown below.

a) Transformation for Integer data type

Various data types for storing the integer values are mapped to a single key ie 0 for showing the equivalent compatibility between the similar data types.

$$\left.\begin{matrix} int \\ integer \\ smallint \\ bigint \end{matrix}\right\} \rightarrow key"0"$$

b) Transformation for Character data type

Various data types for storing the character values are mapped to a single key ie 1 for showing the equivalent compatibility between the similar data types.

$$\left.\begin{matrix} char \\ varchar \\ blob \\ text \end{matrix}\right\} \rightarrow key"1"$$

c) Transformation for Date data type

Various data types for storing the date values are mapped to a single key ie 2 for showing the equivalent compatibility between the similar data types.

$$\left.\begin{matrix} date \\ datetime \end{matrix}\right\} \rightarrow key"2"$$

d) Transformation for Integer, Int and Real to other data type

The Integer, Int and Real data type on the source side can be saved as string or text data type on the destination side. This transformation does not lead to loss of data. It may require type casting in future for the defined mathematical or other operations. The mapping function is as shown below.

$$integer \rightarrow \begin{cases} char \\ varchar \\ string \\ varchar2 \end{cases} \quad int \rightarrow \begin{cases} char \\ varchar \\ string \\ varchar2 \end{cases} \quad real \rightarrow \begin{cases} char \\ varchar \\ string \\ varchar2 \end{cases}$$

e) Transformation for Date to other data type

The Date data type on the source side can be saved as string or text data type on the destination side. This transformation may not lead to loss of data if the container has the required memory to hold it. It may require type casting in future for the defined mathematical or other operations. The mapping function is as shown below

$$date \rightarrow \begin{cases} string \\ varchar \\ text \\ varchar2 \end{cases}$$

4.3.3 Insertion of data in DDB environment

This is the final process of migration of the selected data obtained from the source to the destination table(s). I have shown the use of more than one table to indicate a DDB environment. The source data gets migrated to more than one table as shown in the Figure 4.7.

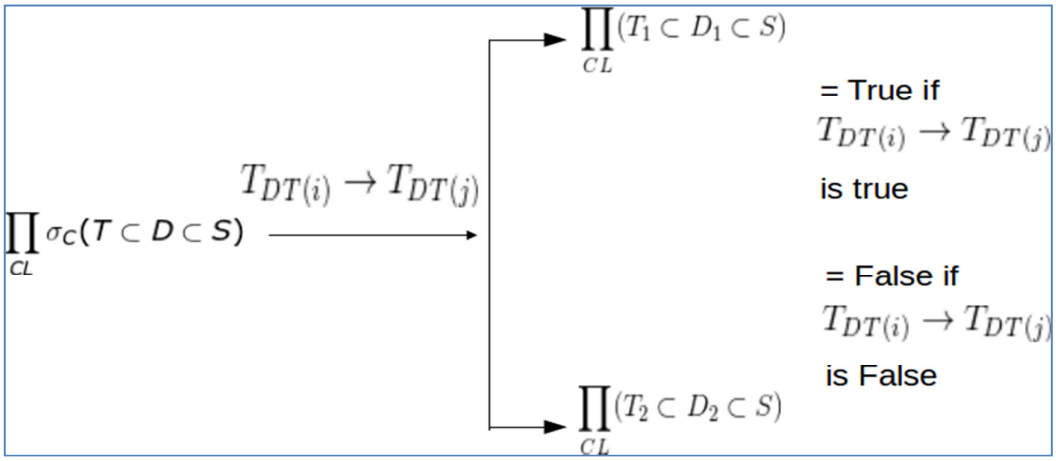


Figure 4.7 Data Migration in DDB environment

As shown in the Figure 4.7, the list of selected attributes from the table and filtered using a condition c, undergo a data transformation from DT (i) to DT (j) for as many tables selected from the database of a given server where DT is the data transformation.

4.4 Situations for HDMT implementation

Various existing DBA tools are used by DBAs for different purposes. The HDMT focuses on the maximum level of details for data migration under the situation that DBAs are able to share some minimum amount of connection information. They need to allow for data communication from the IP address of the tool server. Also, this tool is different from other migration tool in a way that the DBAs can exactly select the data attributes to be migrated and also that it requires ETL operation to the minimum. This service of data exchange provided by HDMT can be most appropriate in the following two situations:

4.4.1 Data Communication Between Inter Connected Organization

When two or more independent organizations but working towards the same benefit, maintain the data and which can be shared, then the HDMT can be used. For example, the colleges are required to send the student data to governing body and also other departments or cells which provide financial relief to the students. In such a situation, the governing body can directly migrate the authenticated student data to the concerned departments or cell or vice versa. The same situation exists for examination data being migrated directly by examination department to the governing body.

4.4.2 Maximum attributes of entities are similar

The HDMT can be used to communicate maximum possible non key attributes for data migration and for further processing. Under this situation too, the appropriate data attributes can be selected for data migration. For example, the examination related performance details of a student can be directly taken from the DB of the concerned institute authority and migrated to the DB of governing body. In this case, only a part of the student information that is related to the examination gets migrated to the governing body DB server.

4.5 HDMT Features

In order to handle the above two situations, HDMT provides the following features.

a) Connection to remote database
b) Pre - Assessing the DBA action
c) Data Type Compatibility Checking
d) Compatible Data Type Conversion
e) Data Migration Report (in some cases it may vary from Pre–Assessment report which may be a result of constraints at the destination side)

Additionally, HDMT also allows for the selection of subset of data depending on an attribute to implement horizontal fragmentation and the selection of multiple destination tables to implement the vertical fragmentation. In order to

implement the above features, HDMT has the following user interface details and users details.

4.5.1 HDMT User Interface

i) Source – The source is the DB server at the college (or company branch) end from where information will be sent to the governing bodies (or company).

ii) Destination – The destination is the DB server at the governing body side (or company). In case of customized tool (as mentioned in 4.7.2), this location will have the connection parameters already specified. I have implemented the tool considering a general purpose tool.

4.5.2 HDMT Users

HDMT can be used in general, by anybody wanting to migrate the data. Online availability of the service also means that internet connection should be available. But specifically, the tool is useful for people working with DBs.

i) DBA – This user is the person at the college end (or company) who is looking after the college level software system data security and maintenance. This user has to provide necessary database login credentials in order to connect to the source (known to the college) and destination (provided by governing body) database servers. The DBA is involved only in the data uploading and does not have any other privilege on the destination side.

ii) GDBA – This user could be a DBA at the governing body side. He is responsible for the data security and maintenance of software system at the governing body side. There is large amount of data involved in such systems as a large number of Indian educational institutes provide their data to them. This data may be distributed on various database servers. So, the GDBA needs to monitor the smooth functioning of all the database servers during any data transfer from various colleges. The GDBA has all privileges on the destination side DB servers and is involved in monitoring the DB servers. There are various monitoring tools available to check the DB performance of

DDB, so HDMT is implemented for DDB data migration where data is sent to multiple locations.

4.6 HDMT Specification

a) The specification of this tool basically involves the input of various DB connection parameters for the data exchange. This is possible only when the concerned organizations are agreeable to take appropriate security precautions and share necessary connection details.
b) The DBAs play a crucial role in ensuring security by defining appropriate user and their roles for the data migration process.
c) The DBAs also need to allow data communication from the IP address or server name on which the HDMT is hosted for usage.
d) HDMT provides facility to map the columns one to one and initiates the services for assessing the DBA action before executing it. The assessment report shows the possibility of data being successfully migrated to the destination as well as the possible reasons (data format, constraints on data or column) in case of unsuccessful migration. HDMT also allows converting the source data format to compatible destination data format wherever possible. After completion of action, HDMT stores and displays the summary of action executed.
For the data type compatibility checking and conversion, the tool refers to a php file containing associated array with the name – value pair mentioned in it.

4.7 HDMT Assumptions

HDMT has been developed to demonstrate the migration of data between heterogeneous data sources. It is based on the following assumptions.

1. The DBAs of involved organizations for the data communication are agreeable to the sharing of some basic information like IP address of database server, type of database server and user name and password. This information is required for the server connection which is finally required for data migration. The source gives the data to be inserted into the destination

tables. In order to handle security issues, the DBAs can work with specific user name having specific role and responsibility.

2. SELECT is a suggestion for the user to select a column or attribute name. It is NOT an attribute in itself. This imposes the restriction that column name cannot be SELECT.

3. The selection of columns is lateral ie source column *n* is mapped to destination column *n* and so on. There could be c1 columns on the source side and c2 columns on the destination side. The tool allows a user to select a subset of the columns from the source to be sent to the destination. For proper mapping, equal number of columns need to be selected from the source and destination side.

4.8 HDMT as a Utility

HDMT can be implemented by companies for their use and by themselves on their own servers or it can be implemented as a generalized tool on a server accessed by people under different policies. The generalized tool option will require the source and destination related information to be filled in entirely for the actual data communication to happen. The inputs required are the IP address of the database server or the server name, type of database, user name and password. After this, the other inputs like the database, tables and columns can be simply selected to be finally sent from source table to destination table. I have considered the generalized tool for the implementation.

4.9 HDMT Pr-requisites

HDMT tool is based on LAMP platform. It needs a LAMP server for its implementation. Also for implementing the heterogeneous database environment, I have considered MySQL (version MySQL 5.5) and PostgreSQL (version Postgresql-server9.1) database servers. These DB drivers have to be installed on the LAMP server for the migration tool related services to connect to various DB servers. Also, the DB servers should allow

for data communication from the migration tool server for which the DBA needs to apply certain environment settings as shown in Figure 4.8. The changes related to database settings have to be applied in the configuration file of the database server. The configuration file for PostgreSQL is postgresql.conf and that for MySQL is my.cnf. Both the files need to be modified for the purpose of allowing communication from the server where HDMT is deployed.

For Postgresql

File name : postgresql.conf

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------
# - Connection Settings -
listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost', '*' = all
                                # (change requires restart)
```

For MySQL

File name : my.cnf

```
# bind-address = 127.0.0.1  # comment this line out
```

Figure 4.8 DBA settings for allowing HDMT to access DB Server

For the development and implementation, I have selected the open source technologies for which the pre requisites are –

Development : Linux Apache Mysql PHP (LAMP), AJAX, JSON, JavaScript, JQUERY. All the technologies have their advantages in web development.

Deployment : Openshift PaaS by RedHat. This is a cloud server for PaaS.

I have also taken into account the cloud service (VPSDime at https://vpsdime.com/) for MySQL server and Postgresql server installed on CentOS Linux 6.5 and webmin version 1.700.

4.10 HDMT Architecture

The HDMT is a three tier architecture software tool which can be accessed from website as a service. It is SOA based and is hosted on LAMP server which can be on a cloud. The various databases that the tool accesses may also be hosted as part of cloud services. The HDMT architecture can be viewed from different aspects as mentioned below.

4.10.1 HDMT Software & Utility View

The DBAs are the users of this tool and can access HDMT through the internet as shown in the Figure 4.9. The DBA / GDBA connects to the data migration service, HDMT available online as a user. When used, the tool connects to the DB servers on proper authorization and authentication details of the source and destination. The figure also shows that the database servers may be maintained on different cloud services but communicate with each other through HDMT for the purpose of heterogeneous data migration in a DDB environment.

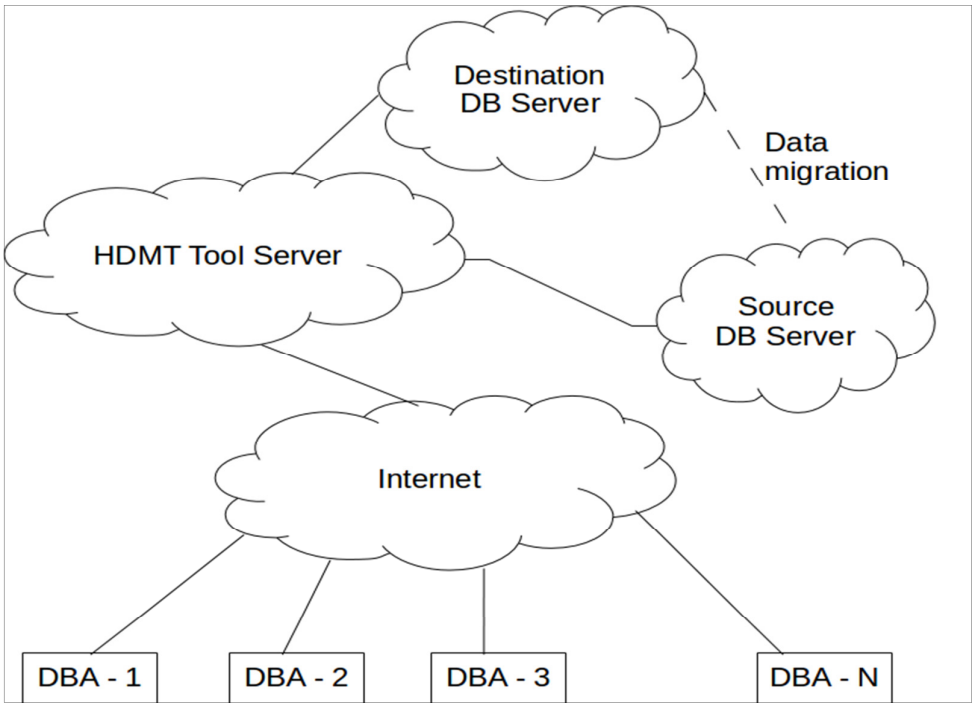


Figure 4.9 HDMT Tool, DB Servers and DBA as users

4.10.2 HDMT Services View

HDMT itself has been implemented as a service available to college DBAs for the purpose of transferring / migrating their data to governing body. The HDMT Tool consists of various services which are called as per the DBA action. These services may in turn call other sub services like transformation and migration so as to complete the DBA action as shown in Figure 4.10. The figure also shows the data migration between heterogeneous database servers in a DDB environment. The DB servers may be implemented and maintained on different cloud servers. The DBA / GDBA can carry out the data migration through HDMT which is itself available online as a service.

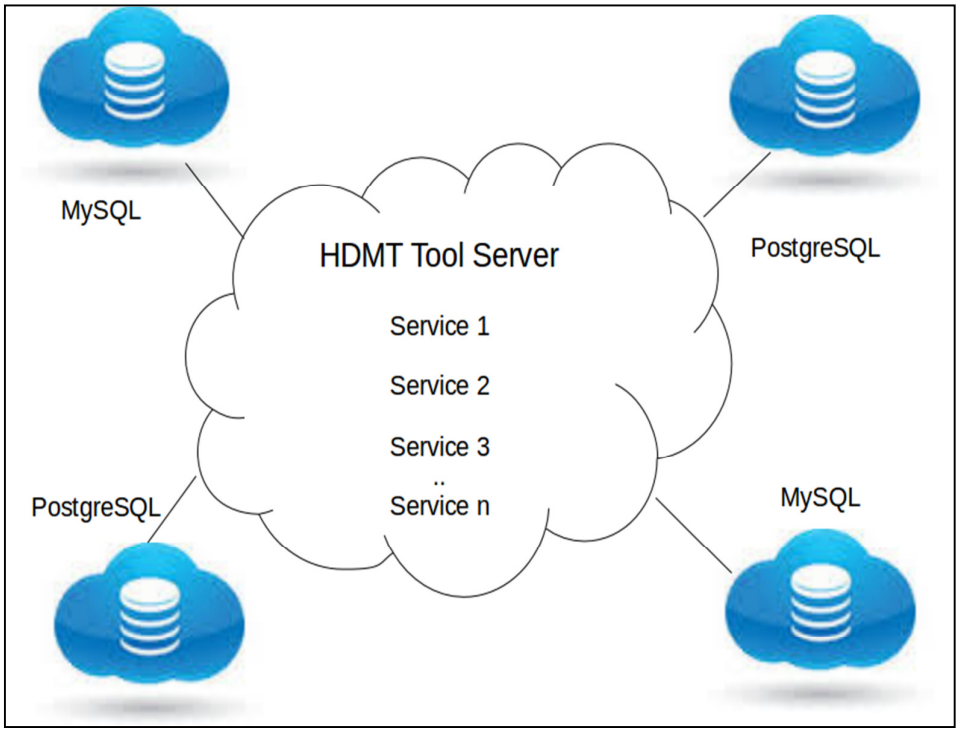


Figure 4.10 HDMT Tool Services

4.10.3 HDMT Three Tier Architecture View

From the point of view of architecture, the HDMT tool can be represented as shown in Figure 4.11. There are a few services shown in the LAMP servers.

These services may in turn call other sub services. I have attached the sample code at the end for reference. The DB servers shown in the Figure 4.11 can be on a LAN or cloud. They must allow data communication to happen from the HDMT tool server. The three tier architecture comprises of the client which works with the presentation layer, the HDMT server which provides the necessary business logic and interface of the data migration tool and the data repository which is used by various database servers to communicate for the purpose of data migration through the HDMT.

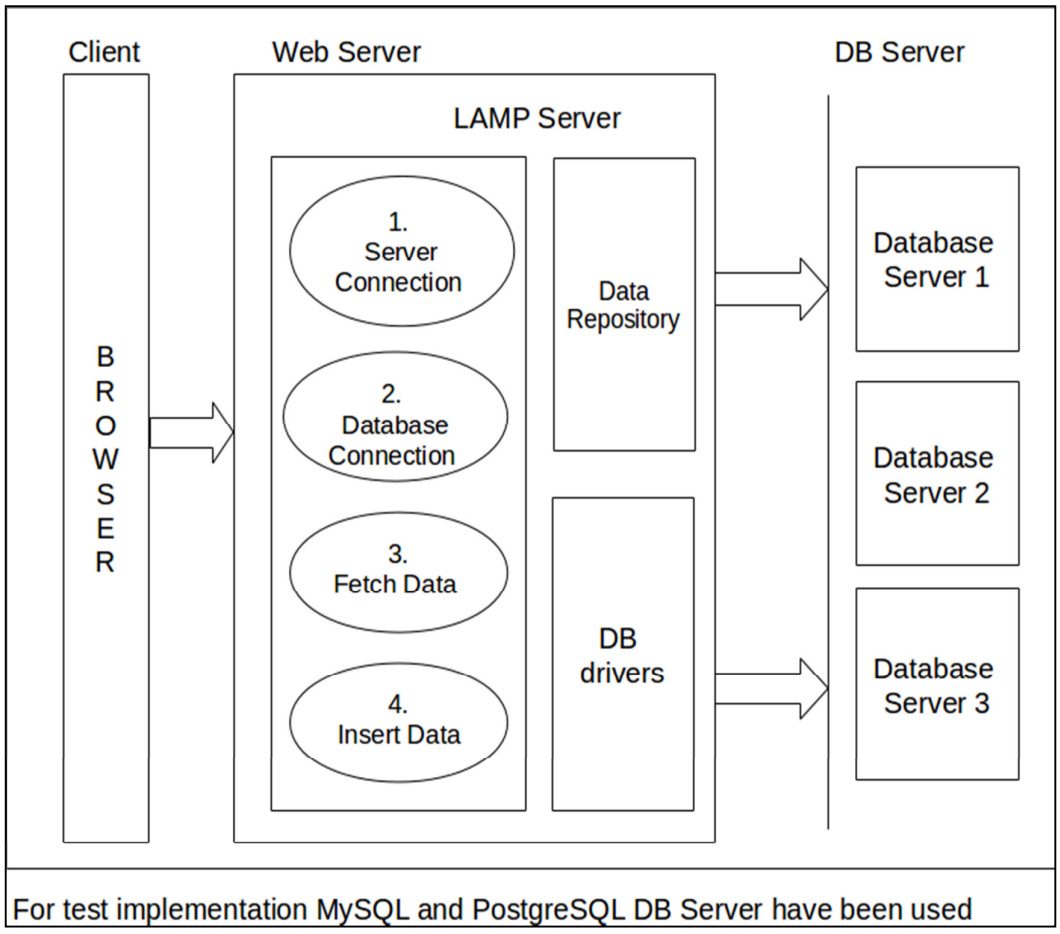


Figure 4.11 Three Tier Architecture based HDMT Tool

4.10.4 HDMT SOA View

The use of cloud services architecture for the tool development and implementation considering the hardware to the application layer can be as

shown in Figure 4.12. Various services like infrastructure as a service, platform as a service, software as a service etc are involved for the development and implementation of the HDMT. Also, HDMT is plugged in with cloud services. The role and responsibility of the data center is maximum at the hardware layer and decreases as we move up the layers. This has been shown with the help of the triangles (standing and upside down).

There are six layers in the tool architecture from the point of view of SOA. The first layer is the hardware layer that is provided by the data center. For the demonstration of HDMT, I have implemented using a cloud service on a rental basis. The data center (openshift PaaS provided by RedHat) provides facility related to the memory and processing units. The second layer is the platform layer which provides the necessary OS and drivers. The environment provided by the data center is as per the client demands. I have used an open source OS (CentOS). The third layer is the application development and deployment layer where there is a need of servers, compilers or software necessary for the developer to work with for software development. The fourth layer is the security layer where anti-virus, firewall, encryption, authentication environment needs to be provided by the data center and developer. The fifth layer is the change management layer which is most useful for any software updations or software change by the developer. For SOA based software system, the focus is on small services to make up a large service. Managing changes in small services or functions which are complete and independent is possible without affecting the overall system functionality. The last layer is the application layer which is the user's layer for working with the system.

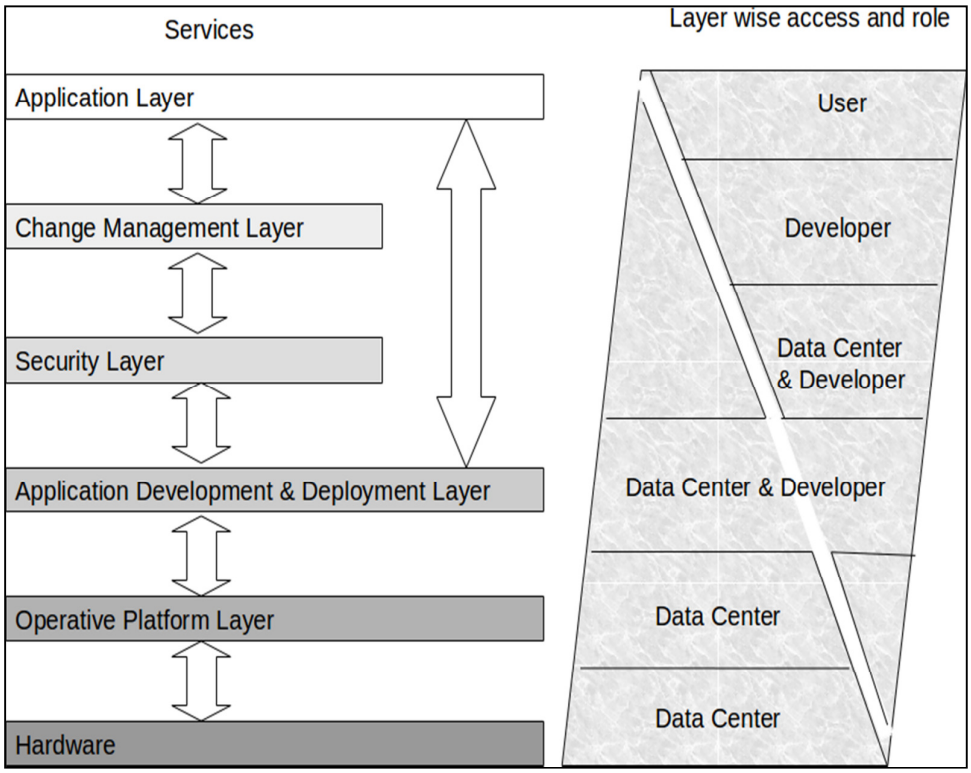


Figure 4.12 Cloud services based tool architecture

The facility provided by the data center depends on the agreement with the clients. The users work with the application layer at the top of services hierarchy as per their role. In HDMT the DBAs are the users of the tool. The DBAs work at the application layer and access HDMT as a service for the data migration in DDB environment. The DBAs need to allow for communication from the HDMT server and then provide the connection and authentication details of the source and destination database for the purpose of data migration.

The service approach for the deployment of HDMT makes the migration service available to all for the purpose of heterogeneous data migration in a DDB environment. The services related to the database also show that the actual data communication takes place at (or between) the data center(s) which is very fast as supported by the underlying hardware and network. All the available cloud services and the service approach, facilitates the job of the DBA in various ways as given below:

- **Location Independence** – The DBA can avail the HDMT service from any location through a web browser and internet connection
- **Migration in DDB environment** – The DBA / GDBA avail the HDMT service for migrating the selected data to multiple destination tables so as to store the data as per the horizontal or vertical fragmentation
- **Heterogeneity** – The DBA is not burdened with the DBMS and can carry out heterogeneous data migration in a DDB environment
- **Reliability** – The DBA is made aware of the pre assessment report before proceeding for the data migration which ensures reliability of the operation

The developer works with the change management and application development and deployment layer as a user of the facilities that are provided by the data center. For example, the data center provides a PHP environment for the developer to use it for software development. The developer and data center play an important role in the security of the system. Few services related to security aspects are provided by the data center whereas the developer also handles the security aspect related to authentication or encryption etc. The hardware and operative platform service is provided and maintained by the data center.

4.11 HDMT Implementation

HDMT is a open source tool and basically requires the LAMP server. The LAMP environment can be implemented on the following depending on the company's needs. For demonstration, I have implemented the LAMP server using cloud services.

4.11.1 Company / Organization Server

This requires a server on which the LAMP environment can be set up. In case of company server implementation, the responsibility of setting up and maintaining the server is with the company. The HDMT tool can be utilized only in case of internal company operations. It will not be accessible online to others outside the company. The security of HDMT is as provided by the web server / LAMP or company policy and standards.

4.11.2 Cloud Server

In general, the HDMT can be implemented on a cloud and accessed through the internet. This service can be accessed as per the terms and policies of the implementer. The security of the HDMT is as provided by the cloud infrastructure or data center.

4.12 HDMT Services

I have worked upon the identification and implementation of following services for SOA based HDMT functionality and its development. The services have been developed using PHP, Javascript, Jquery, AJAX and JSON for client and server side processing. For rich user experience, AJAX technology has been implemented along with JSON that is used primarily to transmit data between a server and web application, as a layer above XML.

a) **Authentication** – I have implemented this functionality as a service to authenticate the credentials based on the user role. The GDBA is the administrator of the tool and will have the highest responsibility. This concept is implemented for the purpose of data exchange between heterogeneous DBs and meta data exchange between web server and the database.

b) **Pre – Assessment service** – This service is implemented to read schema detail and check for possible obstacles that will occur during the execution of query for migration of data.

c) **Data Type Compatibility Checking** – This service shows the data type incompatibility in case of variation in data type between the mapped columns.

d) **Compatible Data Type Conversion** – this service is used to customize the insert query as per the compatibility expected by Destination Database by considering the same as on source database.

e) **Data Migration Reporting** – this service is implemented to show the status of the migration and the number of rows migrated successfully. In case of any database related constraint error, the report will reflect the same.

Continuing the discussion on features further, I have identified a few services and sub services for the implementation of the features. These services along with the user interface are as follows :-

4.12.1 Server Authentication & DB Name Display

This service takes parameters like the IP address, port number, user name and password. The service is implemented individually for distinct DB servers. The LAMP server must have necessary DB driver installed for the implementation and execution of the service. The installation of the drivers is done as per the server specifications.

This service gives the list of DB names as an output that is shown in the control used for displaying the DB names on successful authentication of the DB server. The output ie the list of DB names as shown in Figure 4.13, is in JSON format which is handled by the jquery. In case of unsuccessful authentication, the tool gives an error message as shown in Figure 4.14.

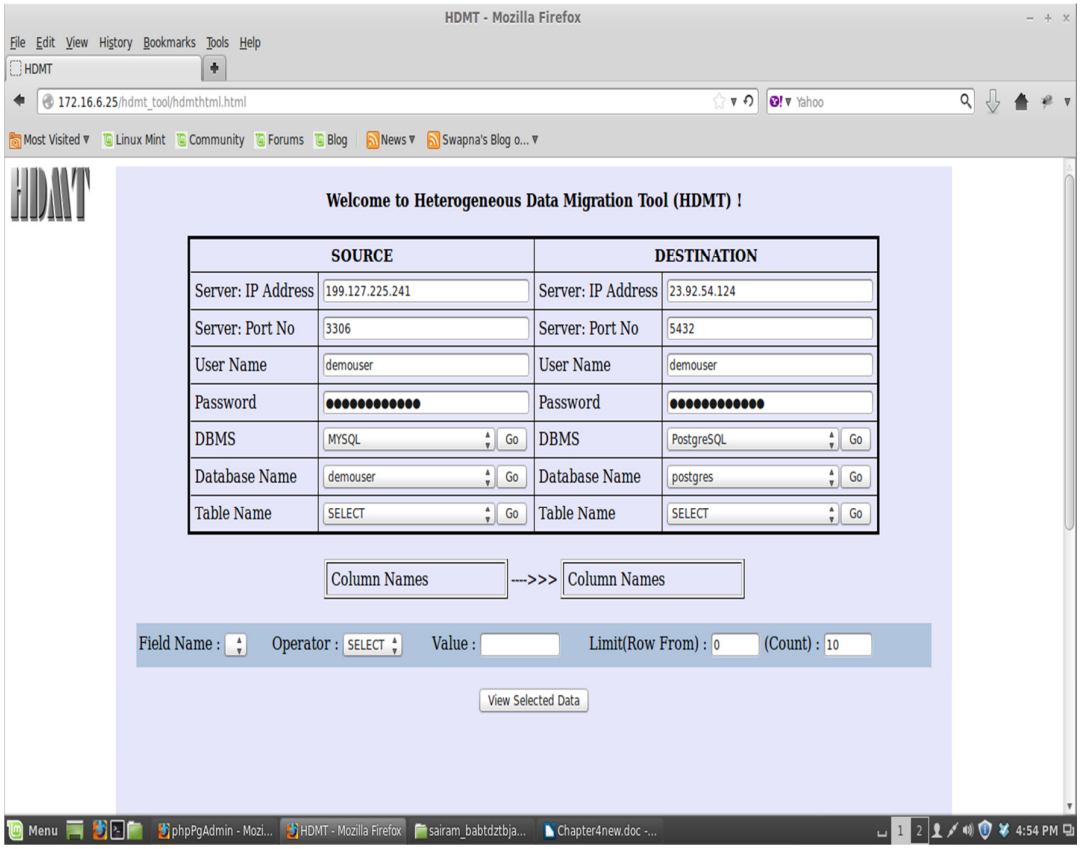


Figure 4.13 HDMT server connection and DB name display

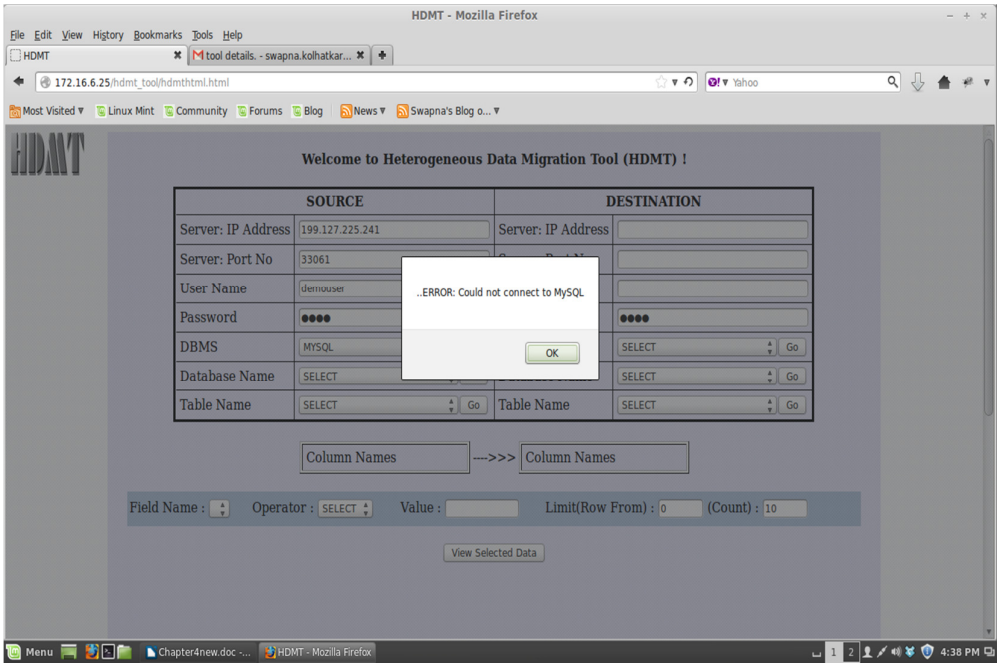


Figure 4.14 HDMT unsuccessful server connection

4.12.2 Database Authentication & Table Name Display

This services takes parameters like the IP address, port number, user name and password and DB name. The server side service is implemented individually for distinct DB servers. This service gives the list of table names as an output that is shown in the control used for displaying the tables on successful authentication of the DB server and DB. The output ie the list of table names, is in JSON format which is handled by the jquery. In case of unsuccessful authentication, the tool gives an error message.

As shown in the Figure 4.15, we are connecting to MySQL and Postgresql DB servers by giving the input details. The Figure also shows a demouser created for the purpose of demonstrating the working of the tool.

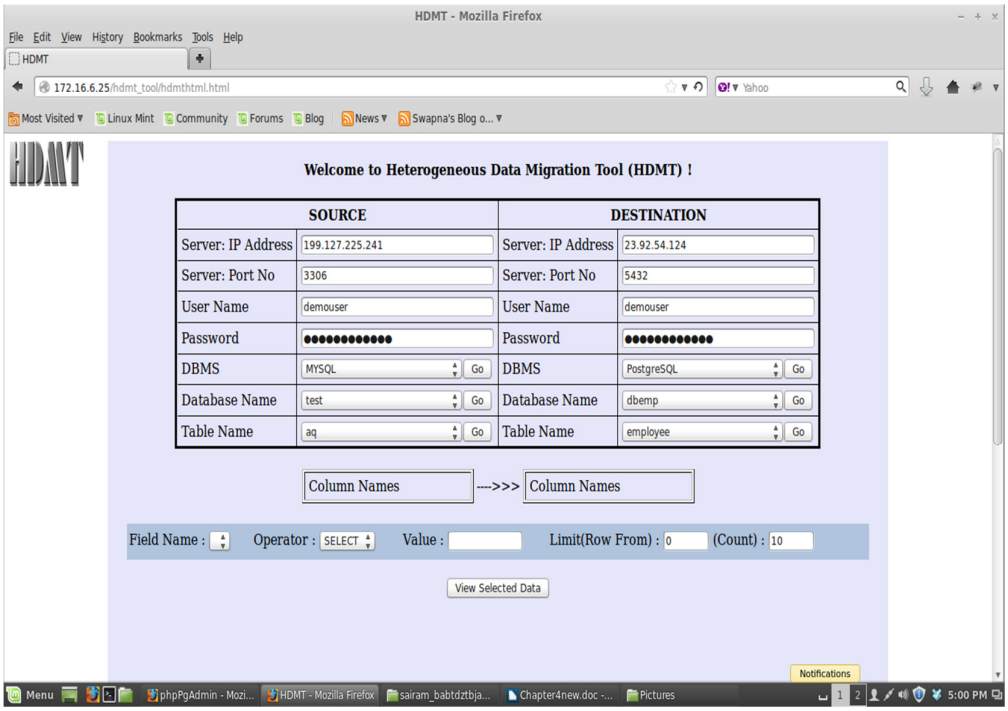


Figure 4.15 HDMT DB connection and table name display

4.12.3 Table Authentication & Column Name Display

This service takes parameters like the IP address, port number, user name and password, DB name and table name. The server side service is implemented individually for distinct DB servers. This service gives the list of column names as an output that is shown in the control used for displaying the columns on successful authentication of the DB server, DB and table as shown in Figure 4.16. The output ie the list of column names is in JSON format which is handled by the jquery. In case of unsuccessful authentication, the tool gives an error message.

The Figure 4.16 shows root as a user for accessing MySQL and Postgresql DB servers for the purpose of heterogeneous data migration. The column names can vary in number for which an independent functionality written in Javascript is called to display the appropriate number of columns. The function dynamically generates the html select control, populating each and every control with the list of column names for the user to select.

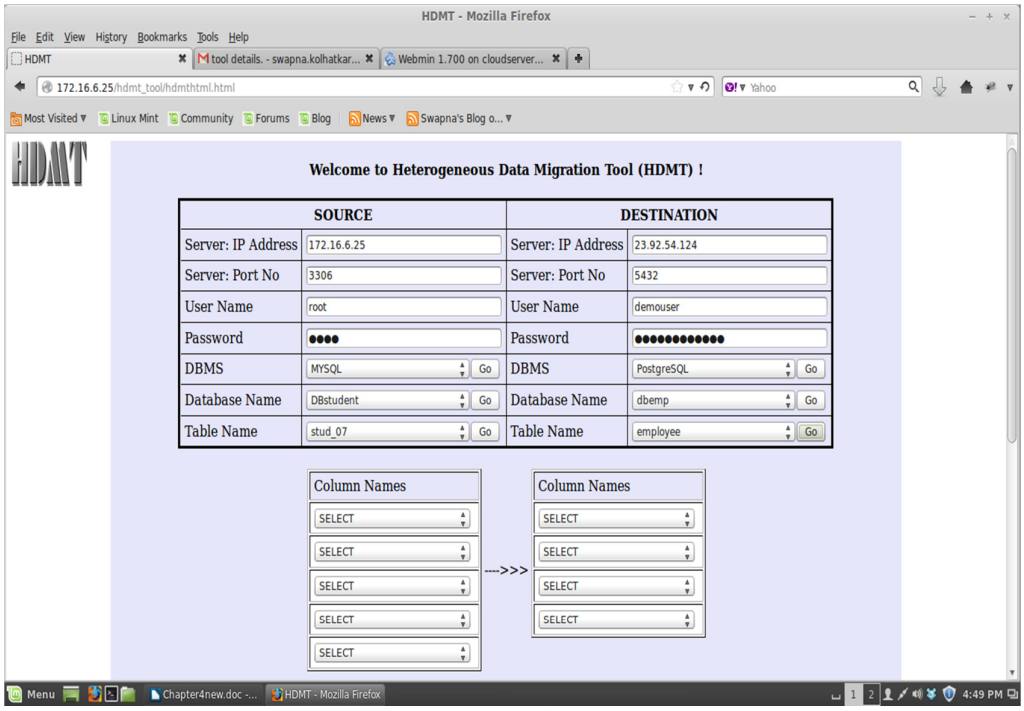


Figure 4.16 HDMT DB connection and column name display

4.12.4 Column Name Selection & Validation

This is a server side process. It composes the selected column names in csv format on the client side and sends the source and destination column name list for further validation. The server side function compares the number of columns on both sides and checks for duplication. In case of problems, appropriate messages are displayed. In case of unequal selection of the number of columns on the source and destination side, following message is shown as shown in Figure 4.17. This checking is done under the assumption mentioned previously for mapping columns.

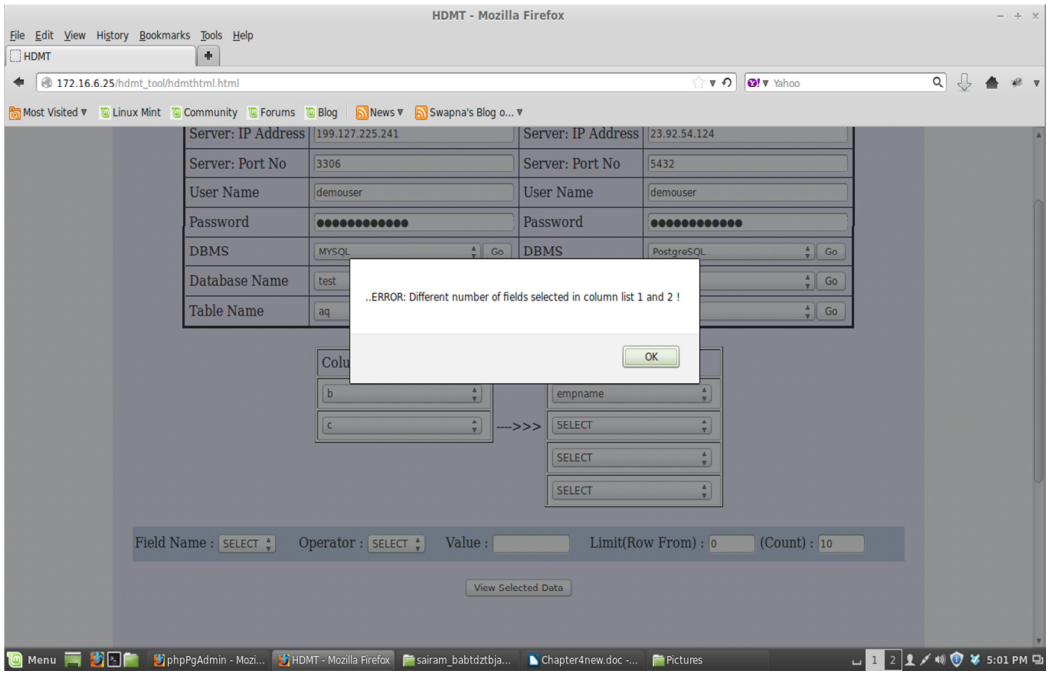


Figure 4.17 HDMT unequal number of column names error display

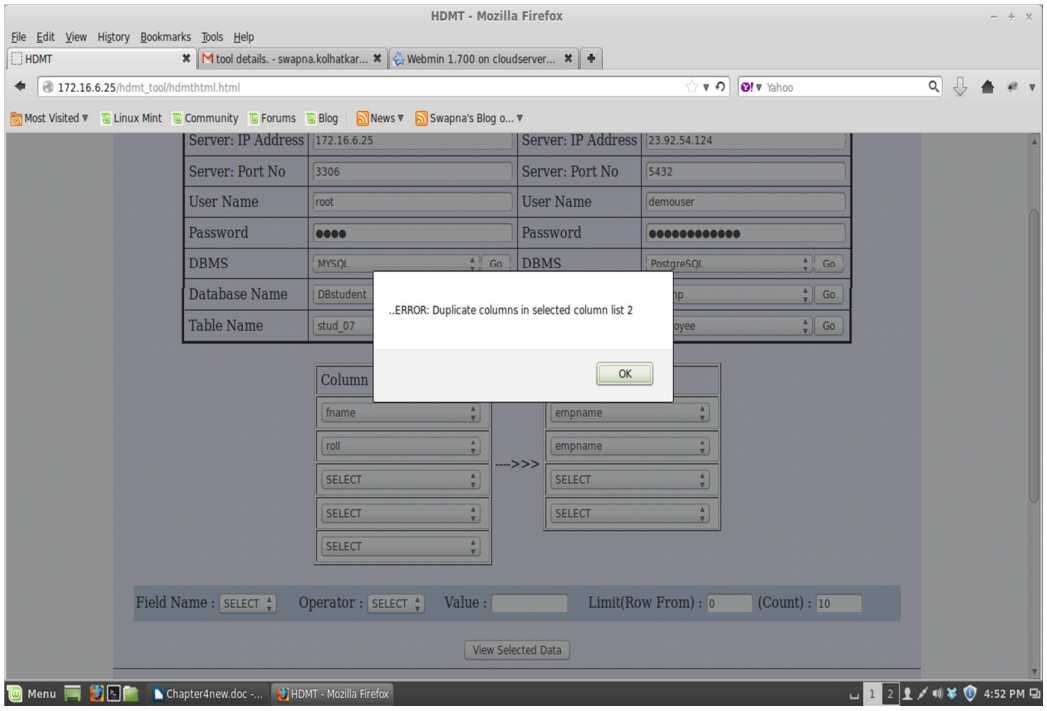


Figure 4.18 HDMT duplicate column selection error message

In case of duplicate column names being selected, the error message is displayed as shown in Figure 4.18. On selecting correct number of columns as shown in Figure 4.19, we proceed for further processing.

The column names may be different as the name difference is immaterial for the data migration process.

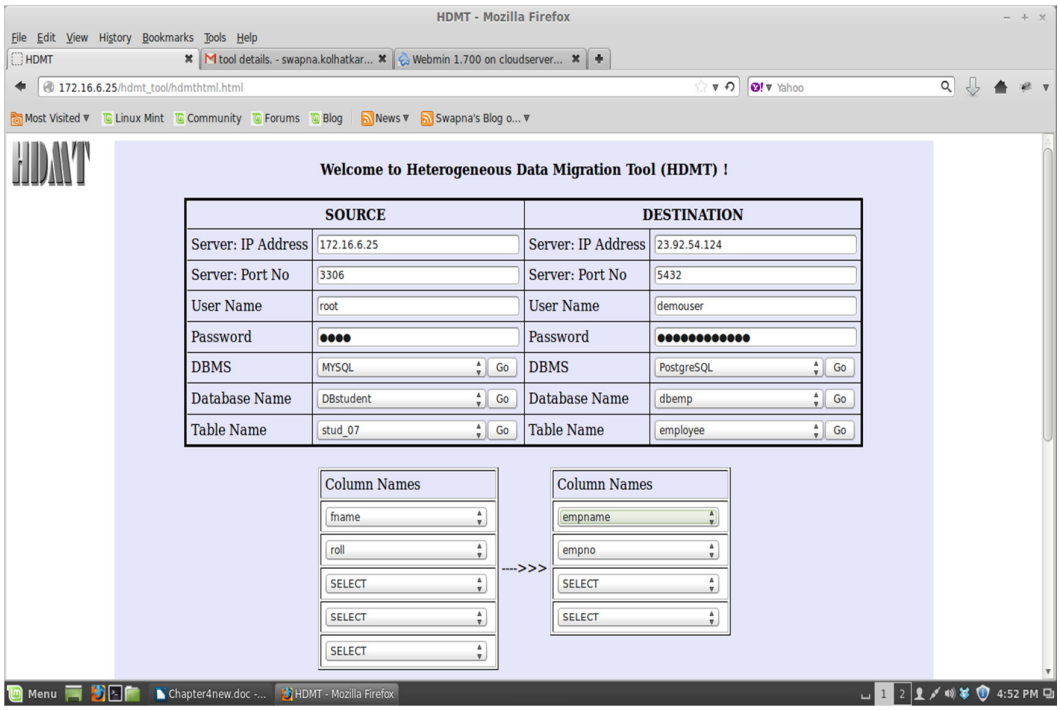


Figure 4.19 HDMT column names selection

4.12.5 Display Selected Data

This is a server side process called on the click of 'view selected data' button and it takes all the connection parameters as input and selects the records from the source side for displaying on the html table. The records sent to the client side are in JSON format which is handled by jquery and sent to the html table. Standard PHP functions have been used to display the data into a suitable format. Various sub services like fetching the data type of the attributes present in the DBA selected list of columns is also called for completing the process of data display. As shown in Figure 4.20, the number of data displayed at a time in the table is as mentioned by the DBA. Here, the

count mentioned is 10 but if kept blank then all the records from the source table is displayed in the web browser to the user. Simultaneously, the other interface related to the migration is made visible for the DBA to decide upon the next course of action.

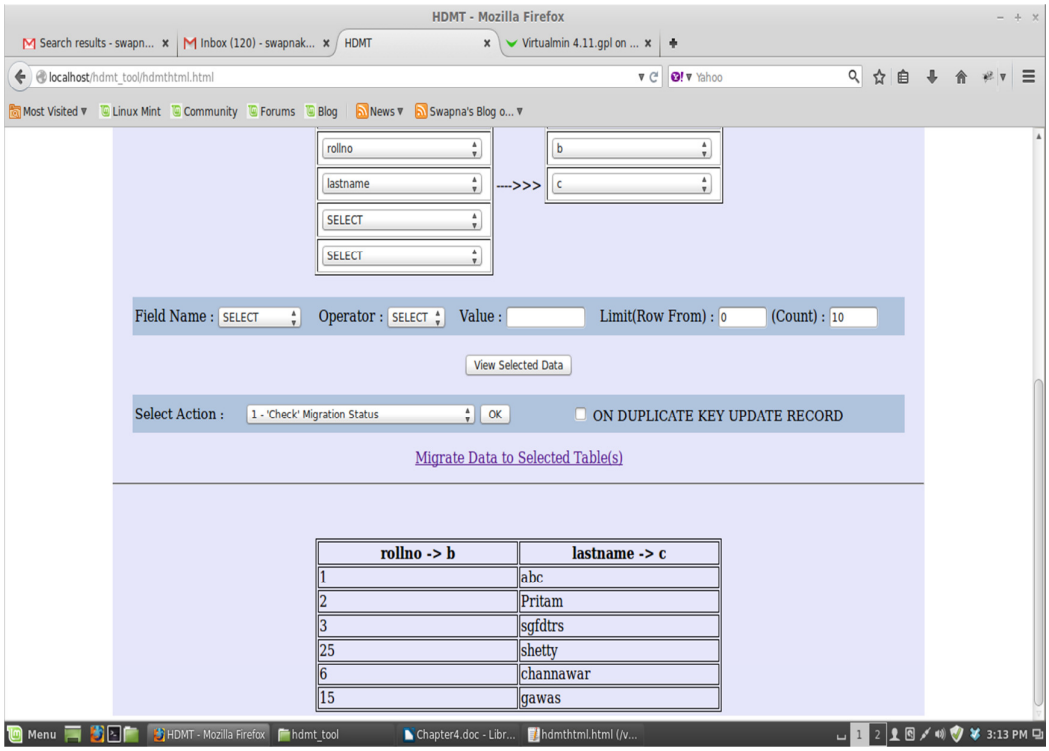


Figure 4.20 HDMT data display

4.12.6 Pr Assessing DBA Action

HDMT provides for some pre assessment of the DBA action ie migration. Before the migration process, the DBA has an option to avail a sub service for deleting any records from the html table by clicking on the record as shown in Figure 4.21. The deletion is done from the html table and not the DB table as shown in Figure 4.23 and Figure 4.24. This deletion may be required for filtering out certain records due to any foreseen problems. The usage of this facility is optional. If all displayed records are to be sent for migration, then deletion is not required. In case the DBA does not delete the record as shown in Figure 4.22, then it is highlighted for the DBA from the reference point of view. The highlighting is just for visibility. The HDMT is a facility that does not disturb the structure or content of the database.

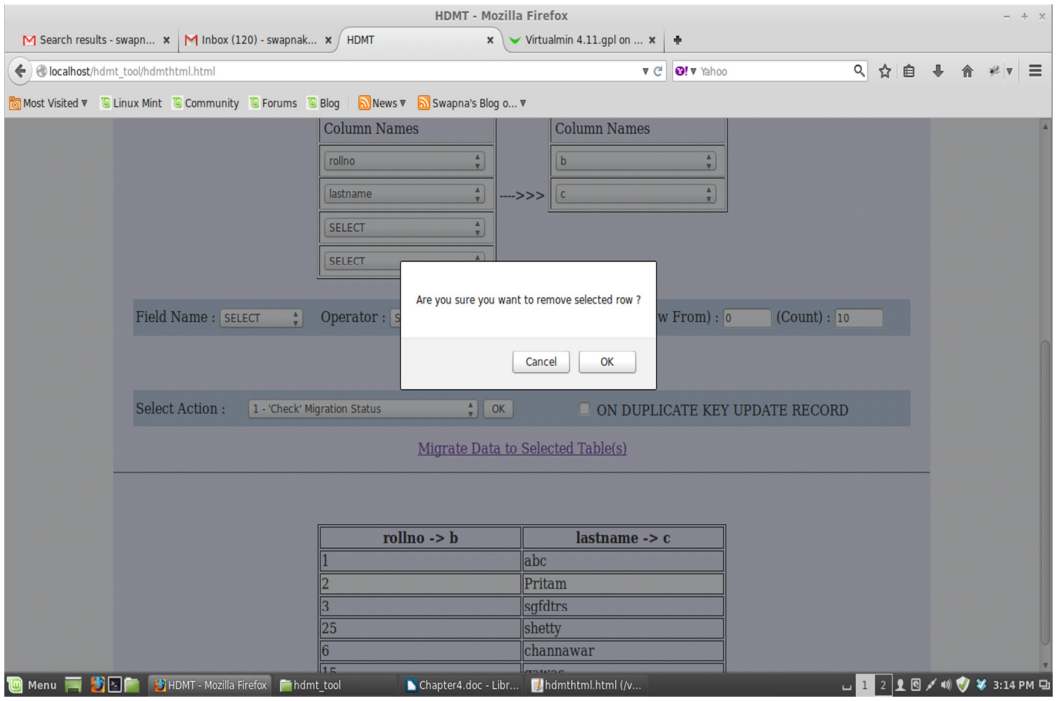


Figure 4.21 HDMT html data deletion facility

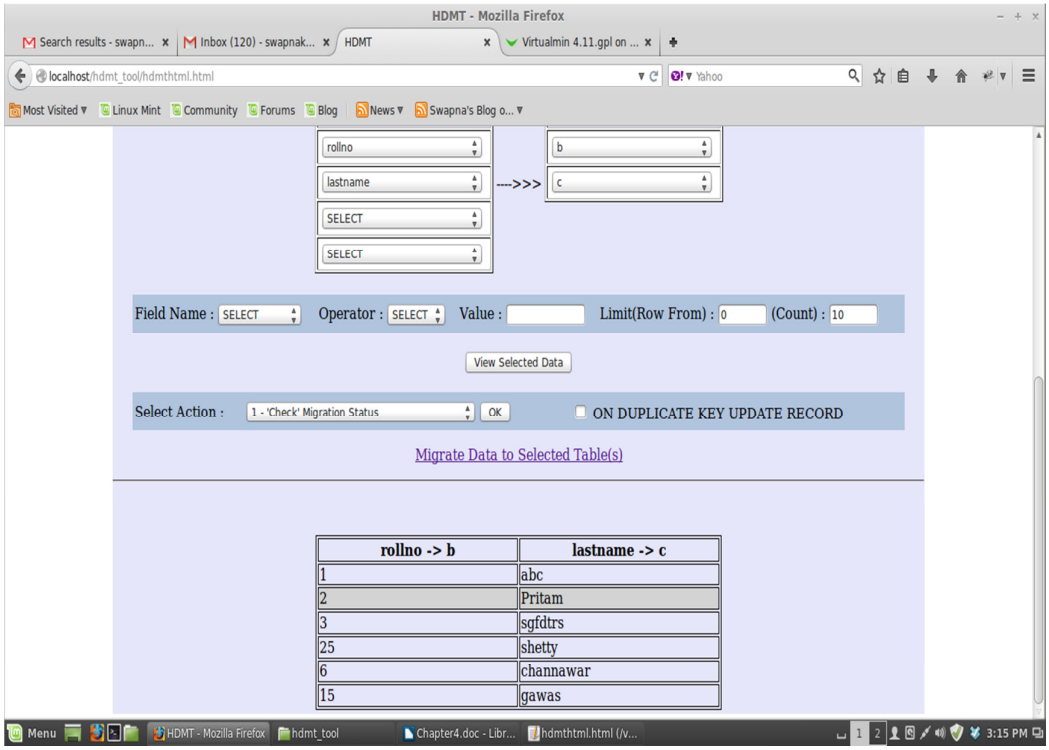


Figure 4.22 HDMT html data deletion facility (data not deleted on selection of 'Cancel')

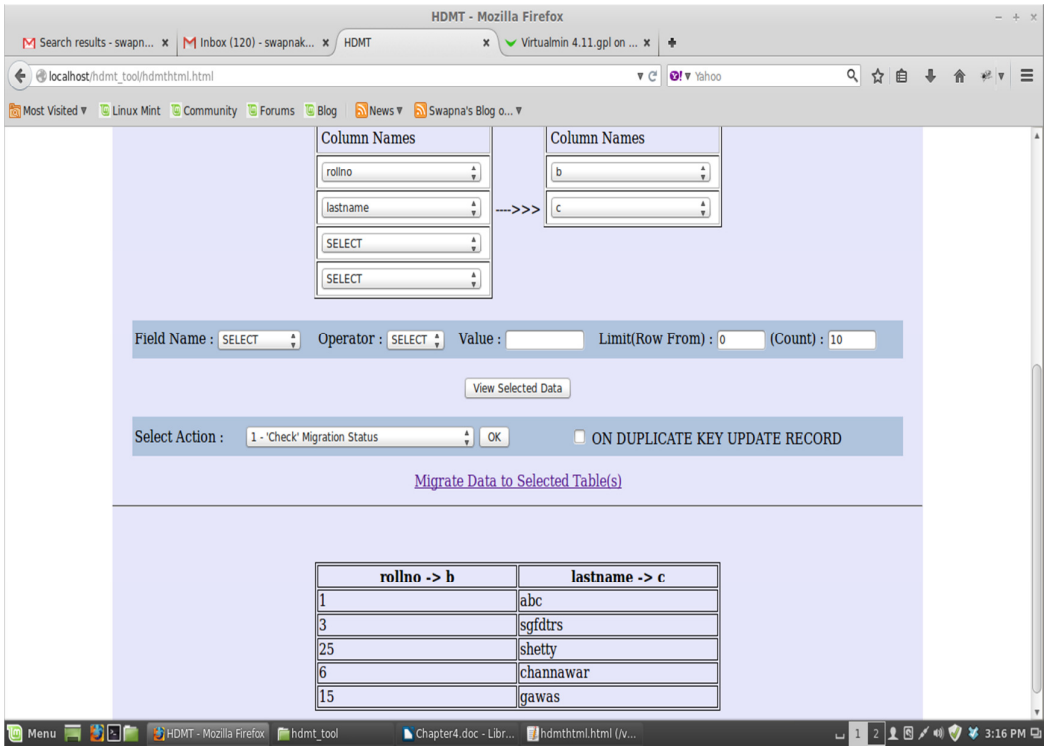


Figure 4.23 HDMT html data deletion facility (data deleted from html table)

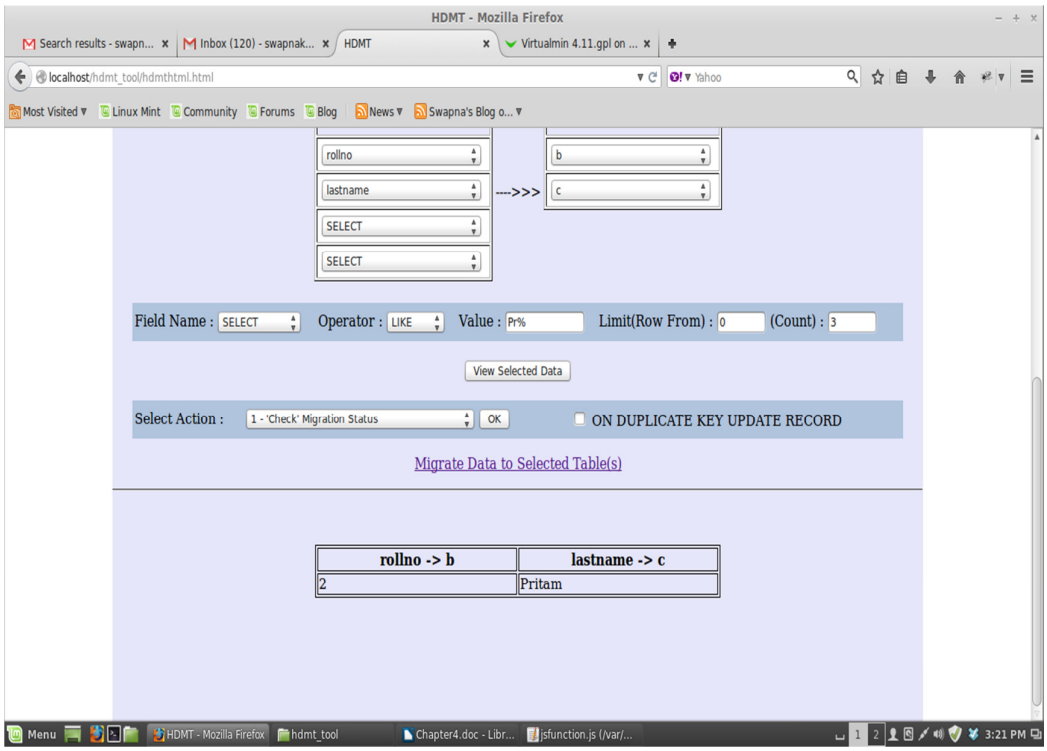


Figure 4.24 HDMT html data deletion facility (data not deleted from DB table)

4.12.7 Data Type Compatibility Checking

This is a server side process written in php which compares the data type of the mapped selected columns between the source and destination. The source and destination has different RDBMS ie MySQL and Postgresql to demonstrate the heterogeneous data migration process. The data type system of MySQL and Postgresql has its differences. These data types are interpreted by PHP to check the data type compatibility.

In case the data type compatibility check is as per the functionality provided by the tool, then the DBA can send the data for migration by selecting the check box for migration. The data type compatibility and convertibility report is as shown in Figure 4.25 and Figure 4.27. The Figures 26, 28 through 31 shows the facility to add other tables for migration in a queue or destroy the queue in case the migration is not to be done for the selected tables added to queue.

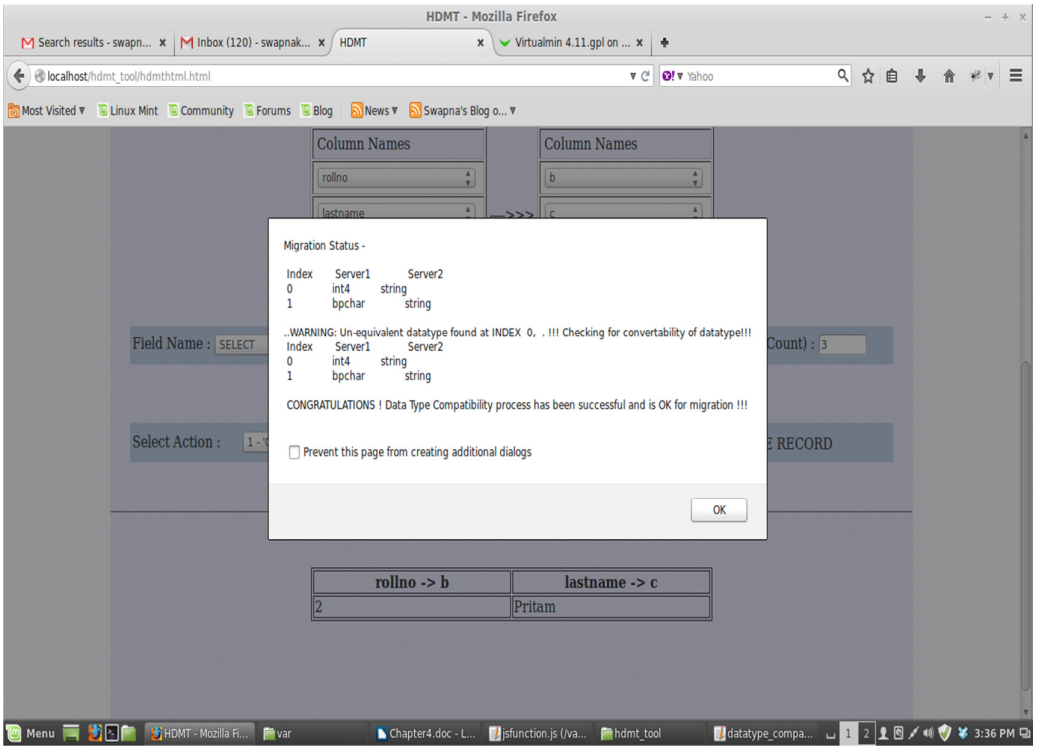


Figure 4.25 HDMT data type compatibility & convertibility checking

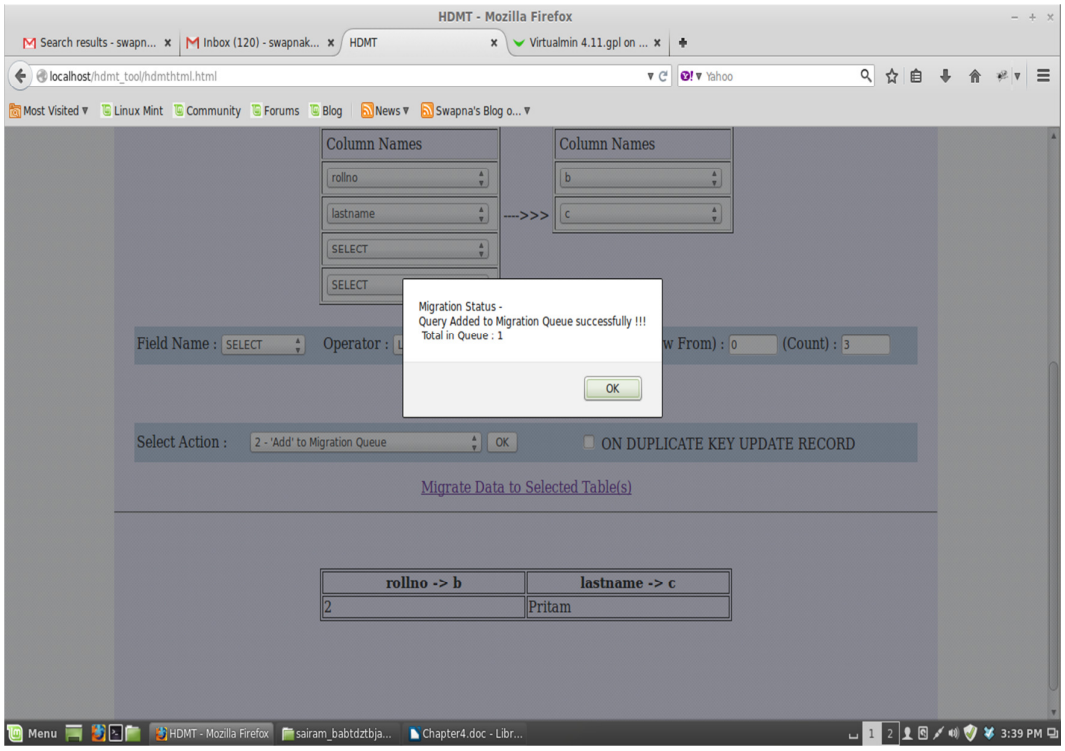


Figure 4.26 HDMT Add Table to queue for migration

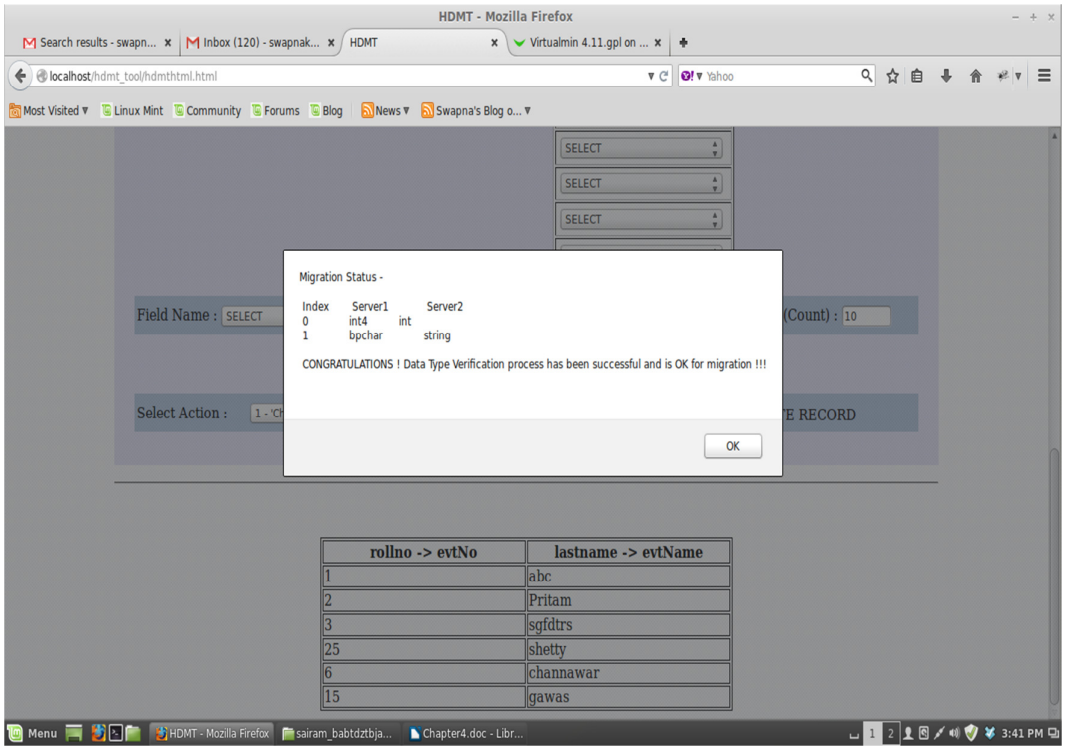


Figure 4.27 HDMT data type compatibility checking for another table

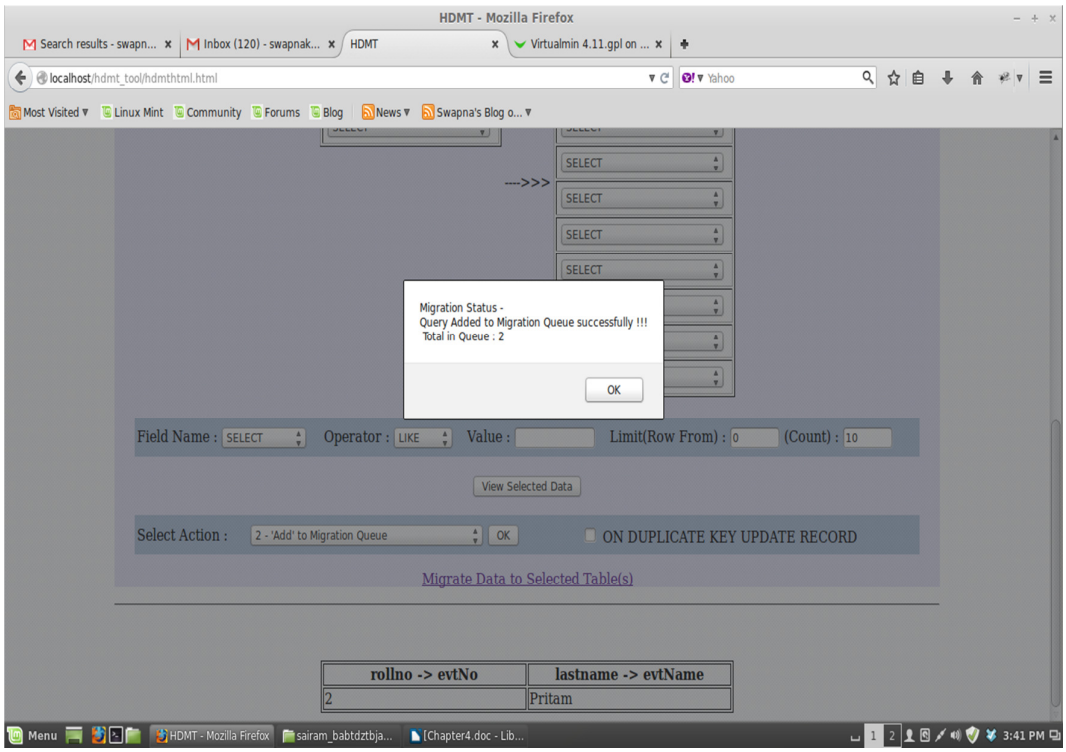


Figure 4.28 HDMT Add another table to queue for migration

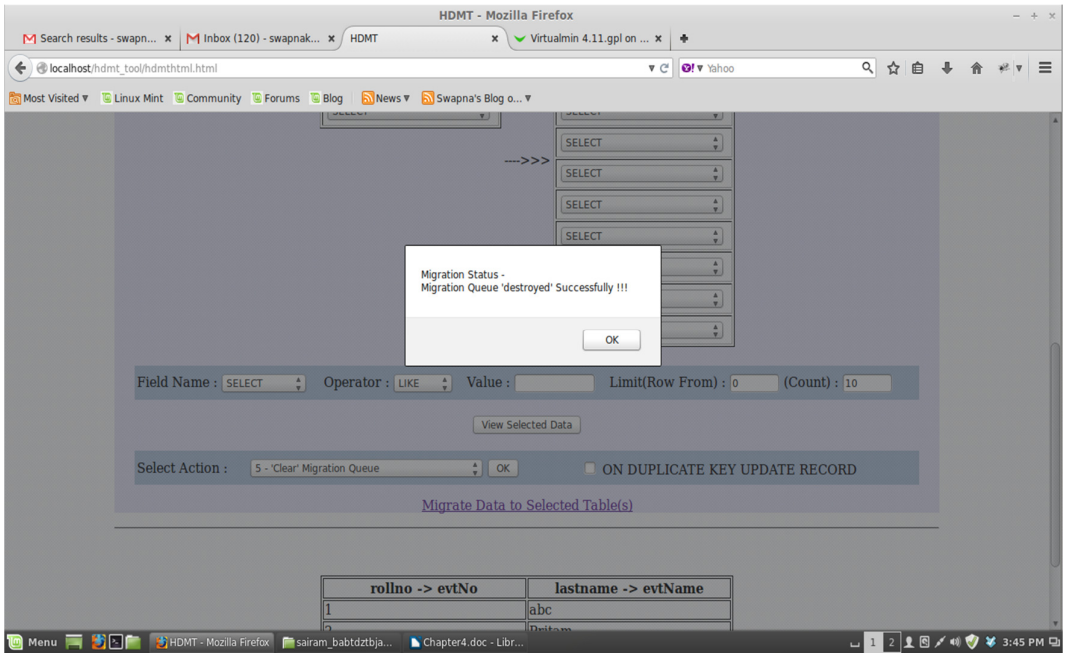


Figure 4.29 HDMT – Destroy the migration queue

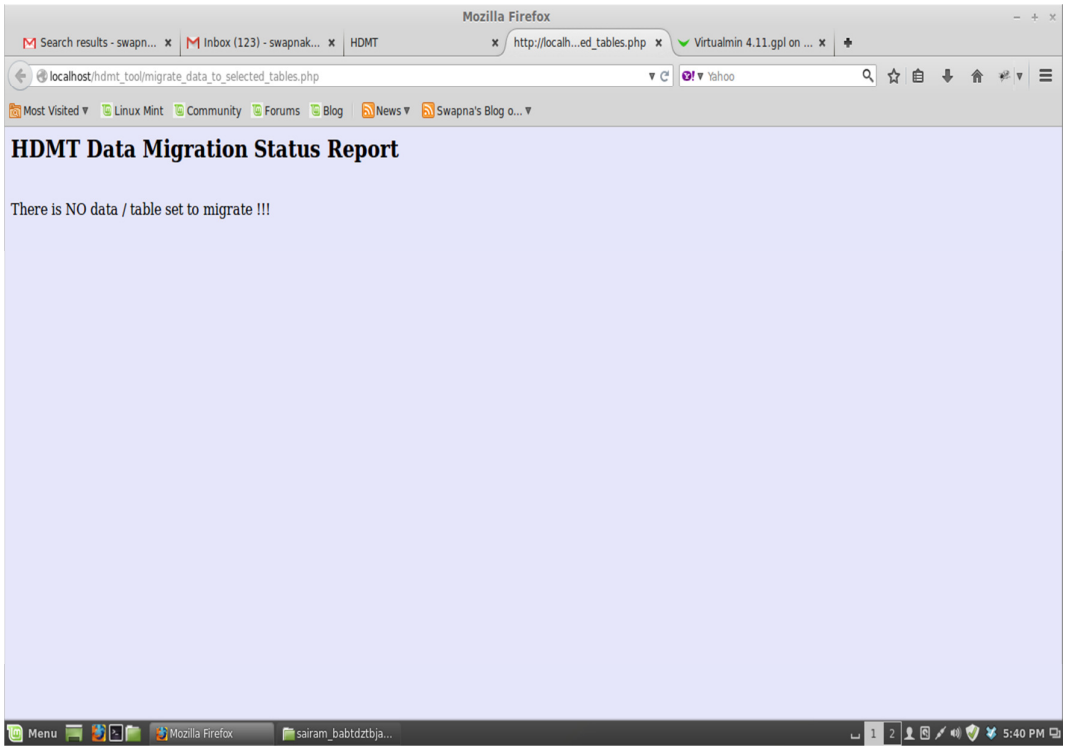


Figure 4.30 HDMT – Data Migration Status Report if no data to be migrated

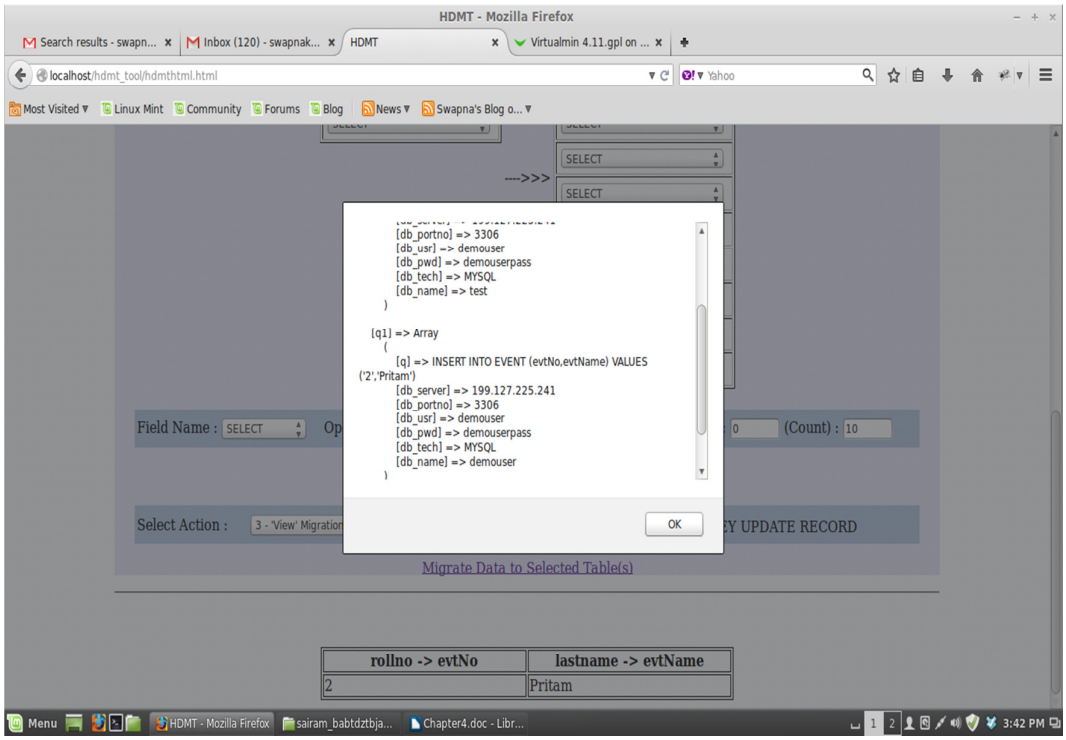


Figure 4.31 HDMT – View the migration queue

4.12.8 Compatible Data Type Conversion

The differences in data type are shown to the DBA. Though the data types are compatible, they may have certain data type differences which restrict the migration of data from source to destination for example small int, int, big int, integer etc. A homogeneous DB would have made the data migration simple due to the similarity in data type, but the same does not hold true for the heterogeneous DB.

The data type compatibility difference in such a heterogeneous environment has been handled with HDMT. It has been handled with the help of a data repository which can be modified appropriately as per the need to handle the wide range of data types, though care must be taken in identification of compatible (equivalent and convertible) data types. The following Figure 4.19 shows a sample code for handling the data type compatibility conversion.

The code uses the associative array of PHP in order to identify the value key pair for defining the compatible data type values as shown below.

```
/* This file is used for providing the data type compatibility information. All the compatible data types will be having the same value and data type as key ie data type name will be key and its value as integer value. This file has been included in hdmtinsertselect.php file for the check_datatype_compatibility() function. */
<?php
$arr_equivalent_datatype_set =
array("integer"=>"0","int"=>"0","number"=>"0","int2"=>"0","int4"=>"0",
"char"=>"1","varchar"=>"1","text"=>"1","string"=>"1","character"=>"1","blob"=>
"1","bpchar"=>"1",  "date"=>"2","datetime"=>"2","year"=>"2", "real"=>"3"
 );
$arr_convertible_datatype_set = array("integer"=>array ("char","string",
"varchar","varchar2"),
"int"=>array("char","string","varchar","varchar2"),
```

```
"real"=>array("text","varchar","varchar2","blob","bpchar"),
"string"=>array("text","varchar","varchar2","blob","bpchar"),
"int4"=>array("char","string","varchar","varchar2"),
"bpchar"=>array("text","varchar","varchar2","blob","string"),
"date"=>array("text","varchar","varchar2","blob","bpchar")
);
?>
```

Figure 4.32 HDMT data type compatibility conversion code

4.12.9 Data Migration

This is the final service provided by HDMT for the purpose of the heterogeneous data migration process for DDB. In case of DDB data migration, the DBA / GDBA can select data to be migrated from the source table and migrate it to more than one table of a DB server on the destination side in a single transaction. This facility of migrating to more than one table in a transaction is required for data that is vertically partitioned in DDB environment. There are four options provided for the DBA / GDBA in order to know the number of tables selected on the destination side for the data migration.

I have used MySQL and PostgreSQL for the demonstration purpose. These DB servers have their differences related to the insert or update of data. The selected records have been successfully migrated between MySQL and Postgresql as shown in Figure 4.33. This Figure shows the execution and completion report of the number of tables in the migration queue.

The DBA can additionally avail the facility of sub service of updating in case of duplicate records related to MySQL DB server.

On again attempting to insert a duplicate set of records into the DB, results in an exception by the DB where the primary key constraint is mentioned as shown in Figure 4.34. The DBA can choose to avail the facility of 'on duplicate key update', but this facility is not supported by Postgresql.

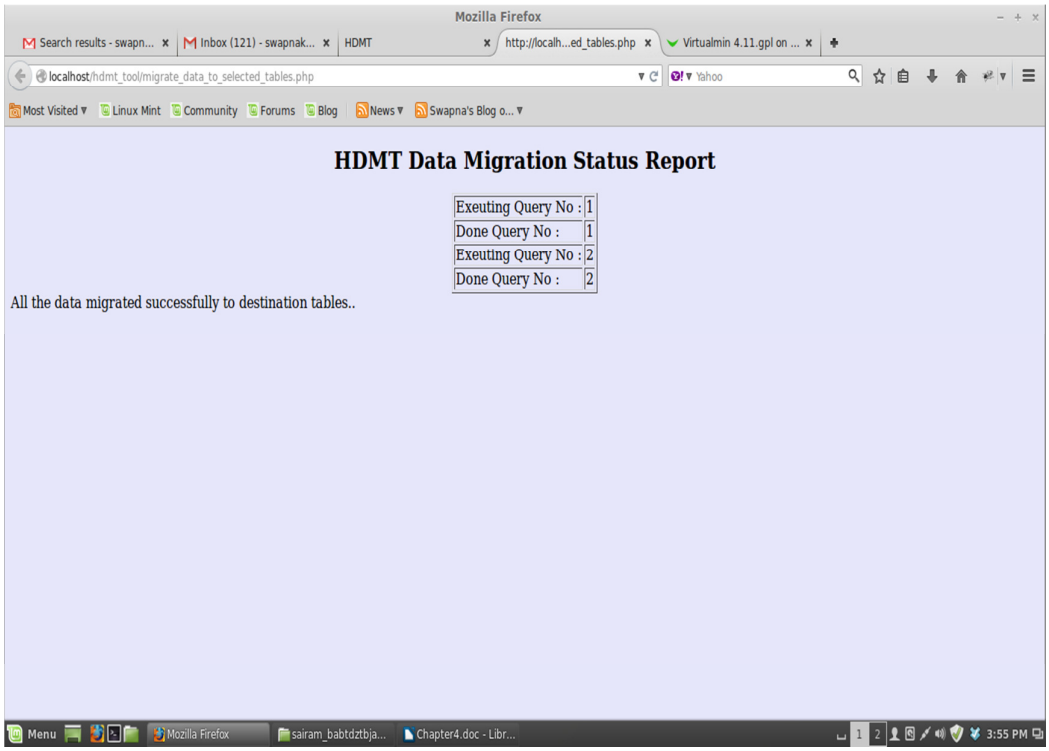


Figure 4.33 HDMT data migration report

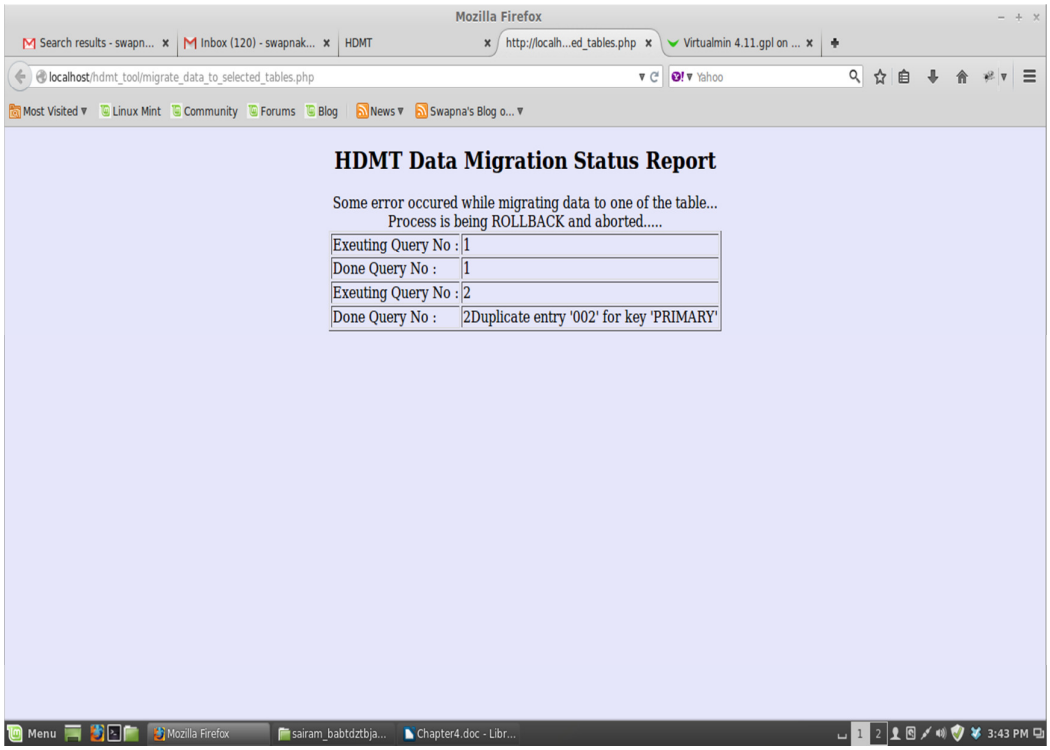


Figure 4.34 HDMT data migration report in case of constraints

On again adding the same set of records to the DB, the DBA gets an exception of duplicate entry where one of the attribute to be migrated is defined as a primary key on the destination side. This exception is applicable only in case of primary key defined on the respective table. In such cases, the data migration report may be different from the pr assessment report as can be seen by comparing the messages of Figure 4.34 and Figure 4.25. The difference in the report is due to constraints defined on the destination DB. In such cases, the destination DB will report the exceptions whenever any rule is violated.

The functionality of HDMT is as shown in all the section 4.12. The purpose of SOA based HDMT is realized along with the use of cloud services.