Chapter 3: Current Database Administration Tools

DB administration is a very important activity for the smooth functioning of the software system. The administrators have a well defined role and responsibility for implementing as well as working with the DB technology. To aid the job of a Data Base Administrator (DBA), a number of tools have been developed to provide necessary assistance and support. The Global Data Base Administrator (GDBA) is a DBA who is having the responsibility of administering all the servers that are a part of the DDB. Various tools have been developed for the GDBA for the purpose of administering DDB servers. As the GDBA is basically a DBA, the tools are known as DBA tools or DB tools. There are a number of DBA tools; involved in DB or DDB administration, having individual features and purpose and are based on their respective architecture. Out of these tools, only the relevant tools are considered for the study purpose. Specifically, this chapter gives a brief idea of the activities and responsibilities of the DBA and GDBA, followed by the differences in DBA and GDBA, in context of DDB. It also mentions important features of some DBA Tools and compares two DB administration tools which have architectural differences. This chapter is important in realizing key features of DBA tools and their benefits along with the architecture of these tools. This study is useful for identifying the requirements of the tool that is the essence of chapter 4.

3.1 Introduction to DBA Activities

The DBA for any application is positioned in middle – top management in an organization. The importance of their role varies according to the complexity and number of DBs in an organization. A DBA is involved in a large number of tasks like design and organization (physical structure, data definition), security (failure conditions, intrusion) and system performance (performance tuning).

To assist the DBA, a large number of tools are available. These tools are related to administration and monitoring tasks which together ensure the proper functioning of the DB. Another important task performed by the DBA is the evaluation and comparison of DBMS's so that the correct product can be selected to meet the DB and customer specifications. Apart from the pricing and support agreements, in general, the DBA should consider technical details like:-

- Data structures supported – The DBA must select the data model to be used. Different models like relational, object oriented, network or hierarchical are available which may be selected after business analysis, data analysis and logical design.
- Performance – Different factors such as software implementation, hardware support, CPU power, main memory, disks required, volume of data etc have to be considered before any DBMS being implemented in the organization.

Along with the technical and economic considerations for the DBA, there are many tools available, in addition to a DBMS, that are required for report generation, data analysis, data migration, data administration, performance monitoring etc for easing the work of DBA.

3.2 DBA v/s GDBA

A DBA is more concerned with the administration of a single server where as a GDBA is having the responsibility of administering all the servers that are a part of the DDB application. The study involves SOA and DDT and hence focuses more on the combination of DBA and GDBA activities for the purpose of developing a software tool. The following information is useful in understanding the differences in the two situations.

3.2.1 DBA Tasks

The job of a DBA as mentioned in most of the Database Administrator's Guide is primarily related to all DB activities as mentioned below.

- Evaluate the Database Server Hardware
- Install the Database Software
- Plan the Database
- Create and Open the Database
- Back Up the Database
- Enroll System Users
- Implement the Database Design
- Back Up the Fully Functional Database
- Tune Database Performance
- Download and Install Patches
- Roll Out to Additional Hosts

3.2.2 GDBA Tasks

I found useful resources with IBM Knowledge Center which have presented excellent contents on the GDBA. The GDBA is defined as a DBA who is additionally assigned the responsibility of planning and setting up a DDB or could be a additionally appointed person who is responsible for the planning and setting up of DDB and following tasks :-

- ***Plan and design*** – Design a DDB that will best meet the software system needs. The needs and goals of the enterprise must be considered when making the decision to use a DDB. Few operations that require particular attention when forming the strategy are networking protocol, system security, job accounting, problem analysis, backup and recovery processes.

- ***Set up a distributed relational DB*** – prepare the application requesters and application servers to send and receive work. When the operating system is installed, DDB support is installed. However, some setup work may be required to make the application requesters and application servers ready to send and receive work.

I had set up a DDB as part of my study. In case of a DDB server environment, the registration of DB servers is done under a particular server group. The DB software provides for the registration of such DDB servers and allows the GDBA to access information related to the DB Servers. The registration can be done either by registering single server at a time or by registering the DB server group file. The majority of these server group files are XML based with the following structure as in Figure 3.1. I had worked on registration up DDB and creating server group file with Microsoft SQL Server Management Studio.

The MS SQL Server related sample .regsrvr file consists of following XML elements :

```
<?xml version="1.0" encoding="utf-8"?>
<registeredservers>
  <servertype id="" name="">
    <server name="">
      <connectioninformation>
        <servertype></servertype>
        <servername></servername>
        <authenticationtype></authenticationtype>
        <username/>
        <password/>
      </connectioninformation>
    </server>
  </servertype>
</registeredservers>
```

Figure 3.1 Sample .regsrvr file

The study of registering DB servers either individually or through server group file led to the implementation of registration service and the service as shown in Figure 3.2 to show the performance of a server in .NET, more specifically ASP.NET and C#.NET. I have published a few papers[1] related to XML and service implementation using C#.NET.

```
protected void BtnShowXML_Click(object sender, EventArgs e)
  {
   ……...
```

[1] The details of the papers published are mentioned in [25] & [26] of bibliography

```
        strfilename = "C:\\Program Files\\Microsoft SQL Server\\swapna.regsrvr";
        XmlDocument rserver = new XmlDocument();
        rserver.Load(strfilename);
        int i = 0;
        XmlNodeList elements =
             rserver.SelectNodes("//Export/ServerType/Server");
            foreach (XmlElement element in elements)
            {
                        string servername = element.GetAttribute("name");
                        txtXMLFile.Text = servername;
                        …………
            }
            myfunction();
    }

    protected void myfunction()
    {
            string strcon = "Data Source=COMP-20\\ADVANCED_DB;Initial
            Catalog=trial_DB;Integrated Security=True";
            SqlConnection sqlConn = new SqlConnection(strcon);
            try
            {
            sqlConn.Open();
                        string strcom = "insert into t_server (server_no, server_type,
                        server_name, server_ip) values (3,'SQLServer','" +
                        DropDownList1.Items[1]+"','')";
            SqlCommand sqlcom = new SqlCommand(strcom,sqlConn);
            sqlcom.ExecuteNonQuery() ;
            sqlConn.Close();
            }
            catch (SqlException se)
            {
                        txtXMLFile.Text = se.Message;
            }
    }
protected void Page_Load(object sender, EventArgs e)
 {
        ConnectionInfoBase coninfo = new SqlConnectionInfo();
       …..........
```

```
        ((SqlConnectionInfo)coninfo).UseIntegratedSecurity = "false";
        TraceServer trace = new TraceServer();
        trace.InitializeAsReader(coninfo, "");
    }

 protected void processPerformance()
    {
        PerformanceCounter   objMemperf   =   new   PerformanceCounter("Memory",   "Available
Bytes");
        PerformanceCounter objProcperf = new PerformanceCounter("System", "Processes");
       PerformanceCounter objComperf = new PerformanceCounter("System", "Threads");
       Label1.Text = string.Format("{0:#,###}", objMemperf.NextValue()) + "Byte";
       Label2.Text = objProcperf.NextValue().ToString();
       Label3.Text = objComperf.NextValue().ToString();
       if (!Page.IsPostBack)
       {
            foreach   (PerformanceCounterCategory   objPer   in   PerformanceCounterCategory.
GetCategories())
            {
                ListBox1.Items.Add(new ListItem(objPer.CategoryName));
            }
        }
    }
```

Figure 3.2 Sample services developed in .NET

- ***Administration and operations tasks*** – The GDBA has to perform routine administration and operation tasks that involve monitoring DB activity, operating remote servers (in special cases), job accounting, auditing etc. In addition, the GDBA is required to communicate with other DBAs in case of any problems related to system or hardware failure, deadlocks, long running queries, job failures etc. In addition, the GDBA must monitor the network, server and DB for improving the performance of the DDB servers.

A study of various monitoring tools was carried out and in my work I also implemented a service in .NET for fetching the performance or load related information with the assumption that the server under study is a dedicated DB server. There are system tables that are used in evaluating the performance of DB server and which can be accessed to get the performance of the DB server. As a result, the entire process of registering individual DB servers and finally fetching their performance information proved useful in understanding the GDBA role and responsibilities.

- ***Communications*** – A GDBA must plan the networking and communication issues as per the software system under consideration so as to take care of the type of information communicated or the number of connections required for the software system or application to function smoothly.

"This aspect of 'type of information communicated' interested me and I started concentrating on data migration and its issues". Another fact that came across was related to data migration and its constraints which requires necessary authorization and authentication for the actual data to be migrated between distinct systems. The Indian Higher Education sector related Information system software gave me an opportunity in understanding the data upload process and its pre-conditions. I focused my attention to the data migration and its issues as discussed in **chapter 4.** At the same time, for getting full advantage of SOA for agile systems, I also worked on applicability of cloud and its services. This study led to the identification of problem of data exchange between distinct organizations and their distinct system implementation. Few papers[2] have been published with respect to higher technical education related software system and cloud services.

- ***Security*** – A GDBA is faced with two security issues to resolve. First, system to system protection; and secondly, identification of users at remote sites. Also, the data must be available at all times and be protected from intruders. A GDBA may additionally take care of the performance improvement by fine tuning the situation and resolve problems of intrusion detection, blocking certain IP addresses etc.

[2] The papers are [23] & [24] of bibliography

This requirement of security along with the presence of cloud services, I explored in data services available on the cloud. The study[3] involved various types of cloud services and their use in implementing any user developed service. All these aspects of cloud services and application in education sector along with the above five responsibilities of GDBA gives an idea of the needs and applications of GDBA tool. They are as follows:-

- **Needs** :

A GDBA needs web based monitoring, real time performance data and alerts**.** There are software systems or applications or services available online, alternatively software can be downloaded for the purpose of monitoring and getting performance alerts. A paper[4] was published related to development of services in .NET for registration and monitoring the DB servers.

This paper presents some important issues in the administration of distributed databases and identifying key factors to be considered for these purposes and finally proposing the possibility of adopting SOA approach in developing software as services for the administration of distributed databases. The paper mentions that the software community is facing a lot of challenges to ensure that the software has value added characteristics related to automation of processes and also having organization of the automation in order to keep pace with the demands of the situation. It mentions that a number of applications related to banking, retail, telecommunication, e-governance, automobile, defense etc have been successfully implemented with distributed databases. Administration issues are of vital importance to organizations adopting distributed database technology in most of the above mentioned areas. On the other hand, SOA, in the software industry establishes an architectural model for software engineering that aims to enhance the efficiency, agility, and productivity of an enterprise by positioning services as the primary means through which solution logic is represented in

[3] The details of the study have been mentioned in the article - Applicability Of Cloud Computing In Academia i.e. [24] of bibliography

[4] The details of the paper are published in [26] of bibliography

support of the realization of the strategic goals associated with service-oriented computing.

After the implementation of registration and monitoring aspect as services, I focused on other applicability aspects as discussed below.

- **Applications** :

Various applications are required for GDBA activities like registering servers to make a server group, monitoring individual servers, replicating to a server in case of any problems, migration of data, sending alerts or emails to other DBA etc. I studied a few DBA / GDBA tools aimed at such activities as discussed in following sections and concentrated on the application of **data migration**, though the above GDBA tool related features are present in most of the well known DBA tools as the following study shows.

3.3 Few Features of Some DBA Tools

There have been gradual as well as prominent advancements in the tools related to the DBA activity. Most of the DBs have their own monitoring and administration tool like MySQL Workbench (MySQL), SQL Monitor 4 (MS SQL Server), TOAD (Oracle) etc. They have many features required by DBA and also allow for the creation of registered server groups and their administration.

For understanding the evolution history of such DBA tools, it is important to understand that advancements in DB technology happened because of the demand to meet availability of reliable data anywhere anytime as a result of increasing requirements of expanding businesses and commensurate software developments. This facilitated the growth of DDB for which the technological advancements in DBA tools can be categorized as shown below.

- **Networks** – A DDB system needs a high speed network for smooth functioning. The network is independent of the geographical components of the system. With advanced networking systems, the DDB system has been implemented more effectively. At the same time, the need for monitoring and administration of DDB has increased due to the demand for availability of reliable data in addition to problems related to server congestion or network traffic.

- **Intranet based administration tool** – The need to support a variety of DBs, led to the development of a DBA software tool that could be downloaded and installed on a company designated server of its premises. The tool would identify and register the various heterogeneous DDBs that were part of the server groups and would then be used for administration and monitoring activity.

- **Online monitoring tool** – With the focus being shifted from maintaining software infrastructure at the company location to the use of data centers or cloud services, the online monitoring tool was developed. This idea was developed to provide all the benefits of online software utility. This facility combined with the applicability of cloud services, has many benefits.

The above three categories along with the main focus of study ie DDB (homogeneous and heterogeneous) system, gives the below mentioned types of DDB management tools.

3.3.1 Intranet based Homogeneous DDB Management Tool

The homogeneous DDB refers to the fact that only one DB Technology (DBT) is considered. The administration and monitoring of such a DDB system requires LAN connection for identifying and finally registering the independent individual DB servers for the purpose of administering and monitoring them.

3.3.2 Intranet based Heterogeneous DDB Management Tool

The Intranet based heterogeneous DB management tool was developed with an intention to manage and administer data and data servers, centrally. The tool had to be installed on a server and data servers had to be registered for the purpose of data manipulation, monitoring and administration.

3.3.3 Web based DDB Management Tool

The next advancement i.e. the web based heterogeneous DDB management tool was developed to meet the ever growing demands of DB administration using a tool that is hosted online. These tools have been developed with their individual architectures and methodologies.

3.3.4 Few DBA Tools

Based on the broad categories of DDB management tools, there are many tools available in the market, some of which are mentioned below. These tools are meant for the DBA to work on various DDB administrative aspects which can be local or global to the DBA / GDBA. I have discussed these tools with respect to their special features, but concentrated more on the data migration and architecture aspect.

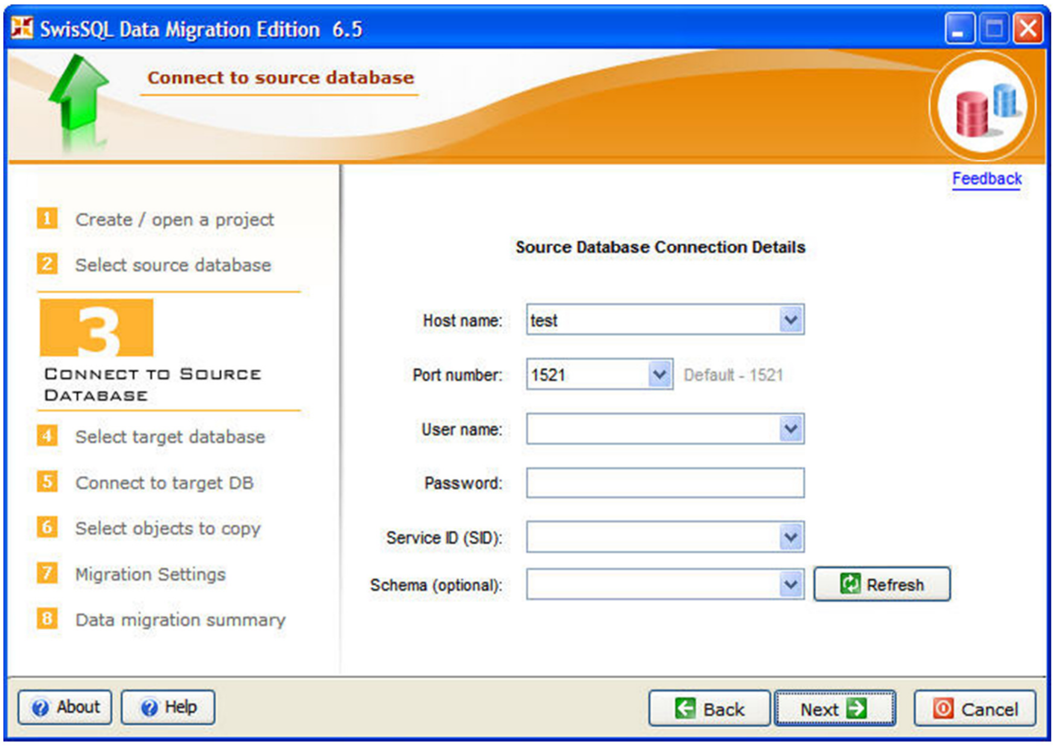
3.3.4.1 Toad

Toad is a powerful and robust tool for the DBA. It provides tool sets designed to manage all data sources and keep DB professionals productive. Toad has different product or plug ins for various DBs though it does not support any non-Microsoft environment. It gives a facility to browse and manage schema and objects inside schema.

The Toad StatsPack Browser (DBA module required) takes beginning and ending snapshots of DB performance statistics and then offers graphical trending and time-series analyses based on those stats. The result is a gallery of useful charts for interpreting DB performance. Toad has a batch mode interface which scans the code and highlights the line that needs attention for DBA.

TOAD offers an advanced security management system and the DBA can use TOAD Group Policy Manager as a central point for the DB administration to navigate to every DB simultaneously, and drill down into any of them to administer DB and schema objects and also perform DB Health Check on the selected DB.

It provides DB performance testing tool that allows DBA to conduct DB workload replay, industry-standard benchmark testing, and scalability testing. It enables the DBA to deploy changes to your DB environment with confidence by reducing the risks associated with patches, upgrades, migrations (as shown in Figure 3.3), and adjustments to virtual machine configurations through the incorporated load testing tools.



Source Database Connection Details

Host name: test

Port number: 1521 Default - 1521

User name:

Password:

Service ID (SID):

Schema (optional):

Figure 3.3 SwisSQL Data Migration Tool

Specifically, DBA can do data migration through Toad Software like SwisSQL which helps the migration and transfer of DB schemas as shown above. The migration done is at the DB level ie entire DB is migrated from source to destination. In case of data migration of more specific entity like tables or records, the DBA has to follow a different route of migration which requires his expertise in fetching the records and verifying them for names and data type information before the actual migration.

3.3.4.2 DbVisualizer :

I have considered DbVisualizer's in understanding the features for better software awareness. Its strength is in visualizing the structure in the DB, its

DB objects and most importantly, table data. DB objects are simple to create using powerful actions in DbVisualizer requiring no manual SQL to be written. DbVisualizer runs in the background so that Admin can work with other tasks. All active background tasks are monitored in a task manager with interrupt support by the DBA.

DbVisualizer is based on the JDBC toolkit. DbVisualizer supports connecting to local, remote and embedded DBs. Connection management is flexible, with options to set up a connection via the step-by-step connection wizard or by manually defining the connection data. It supports a number of DBs and runs on Windows, Mac OS X, Linux/UNIX.

The DB Object Management gives facility to browse the DB objects and administering objects such as tables, procedures, functions, triggers, packages, modules, etc that is it gives DBA visual support to create, alter and modify characteristics for DB objects. The DB Browser is Tree based navigation through DB objects. The Table Data Management is a facility to support for editing table data including binary/BLOB and CLOB data types.

3.3.4.3 AnySQL Maestro

AnySQL Maestro Group offers powerful DB administration, development and management tools for DBs like MySQL, Oracle, SQL Server, DB2, SQL Anywhere, PostgreSQL, SQLite, Firebird and MaxDB. AnySQL Maestro provides all the tools DBA needs to manage and navigate DB objects. DB Explorer is the basic feature of AnySQL Maestro which allows DBA to perform practically all necessary operations upon DBs and their objects. The Object Manager is a tool for operating on DB objects designed as an extension of the DB Explorer with advanced features, such as multiple selecting of objects (for copying, dropping, etc.).

3.3.4.4 SQuirreL SQL

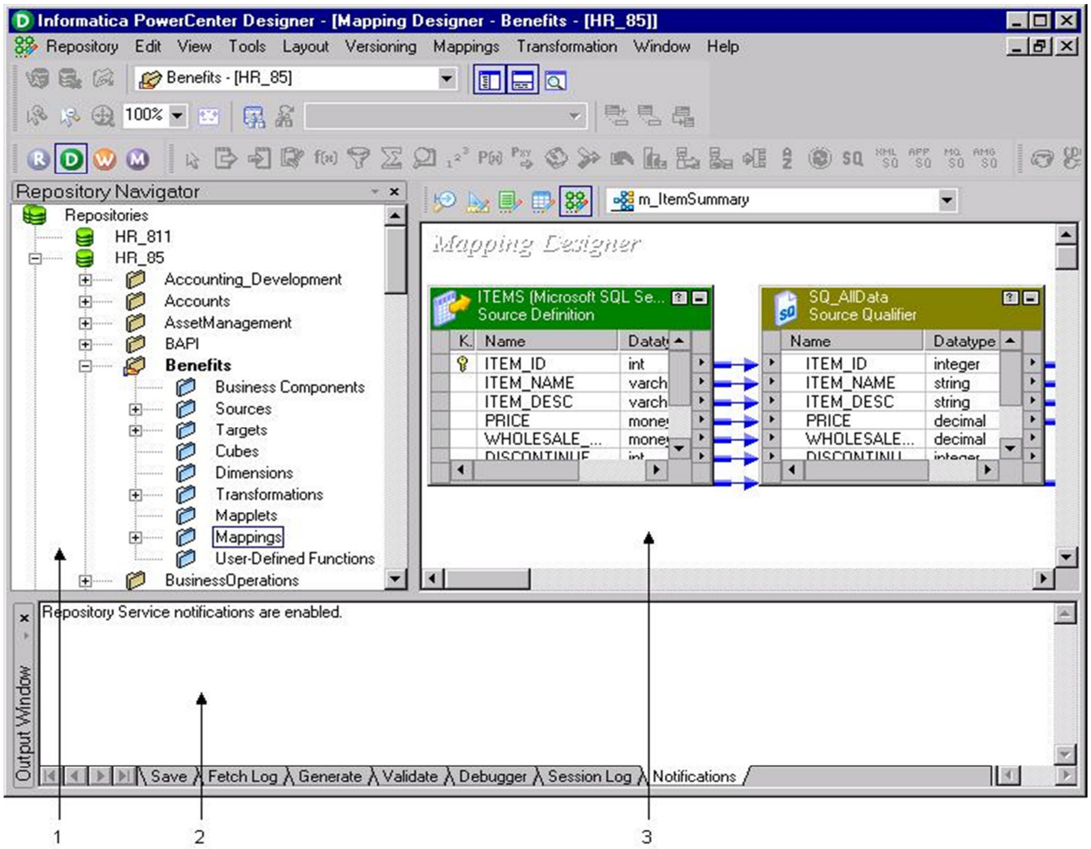
The SQuirreL SQL Client is a database administration tool. It uses JDBC to allow users to explore and interact with databases via a JDBC driver. It provides an editor that offers code completion and syntax highlighting for standard SQL. It also provides a plugin architecture that allows plugin writers to modify much of the application's behavior to provide database-specific functionality or features that are database-independent. As this desktop application is written entirely in Java with Swing UI components, it runs on any platform that has a JVM.

It is free as open source software that is distributed under the GNU Lesser General Public License. It provides a simple graphical interface to relational DBs. it is built using Java, it can access any JDBC – compliant DB running on any machine, allowing remote access to multiple DBs. A SQuirreL user can easily view and edit data in any JDBC-compliant DB, view the DB's meta-data, work with multiple DBs on both local and remote machines, use a single, consistent interface to work with different DB engines, and expand the tool's capabilities and include DB-specific functionality using plugins.

3.3.4.5 ETL tools

ETL tools follow ETL process to extract data, mostly from different types of systems, transform it into a structure which is more appropriate for reporting and analysis and finally load it into the DB and or cube(s). The data is extracted from different internal and external sources. The data is put in a staging area (SA), usually with the same structure as the source. Once the data is available in the SA, it is all on one platform and one DB for further processing ie transform. The transform step of the ETL process, allows users to check on data quality, pivot to another structure and clean the data if necessary. Finally, data is loaded into a data warehouse, usually into fact and dimension tables in the load process. From there the data can be combined, aggregated and loaded into data marts or cubes as is deemed necessary. The entire ETL process is aimed at loading the data into data warehouse for the purpose of data mining. This requires various transformations on the data for better uniformity. I have considered Informatica and Microsoft SQL Server Integration Services in this case to understand their uses.

- **Informatica** is a very good example of ETL tool for extracting the source data and loading it into the target after applying the necessary transformations as shown in Figure 3.4. The entire process of ETL is required for the purpose of a data warehouse and this activity is scheduled to occur regularly as per the business requirements.



K.	Name	Datat
	ITEM_ID	int
	ITEM_NAME	varch
	ITEM_DESC	varch
	PRICE	mone
	WHOLESALE_...	mone
	DISCONTINUE	int

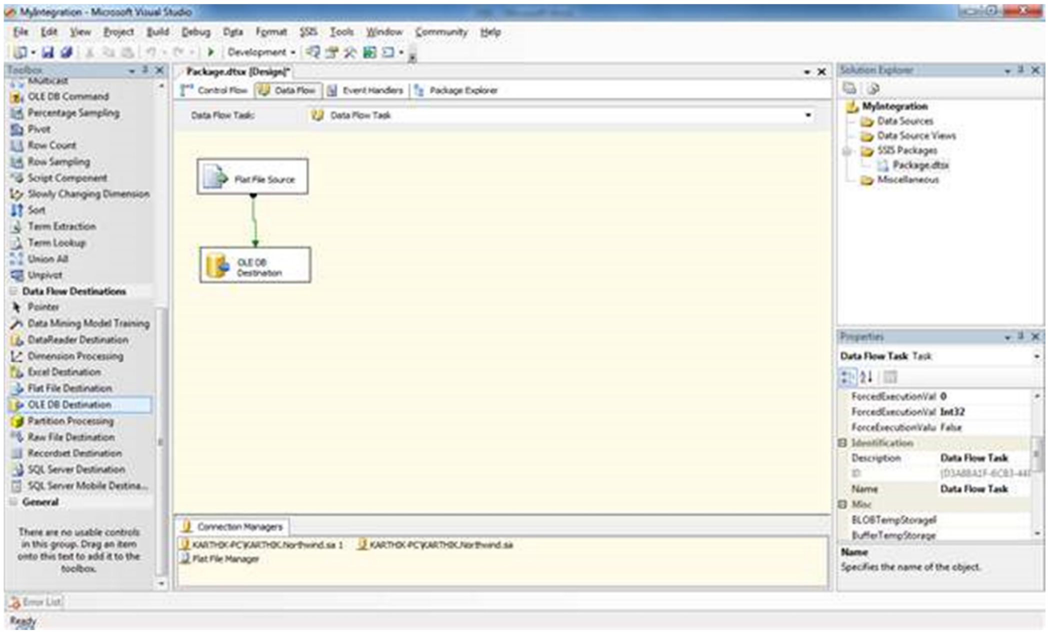
Name	Datatype
ITEM_ID	integer
ITEM_NAME	string
ITEM_DESC	string
PRICE	decimal
WHOLESALE...	decimal
DISCONTINU...	integer

Repository Service notifications are enabled.

1. Navigator 2. Output 3. Workspace

Figure 3.4 Informatica Tool

- **Microsoft SQL Server Integration Services** (SSIS) is a platform for building high performance data integration solutions, including ETL packages for data warehousing as shown in Figure 3.5. SSIS includes graphical tools and wizards for building and debugging packages; tasks for performing workflow functions such as file transfer protocol operations, executing SQL statements, and sending e-mail messages; data sources and destinations for extracting and loading data; transformations for cleaning, aggregating, merging, and copying data; a management service, the Integration Services service for administering package execution and storage; and application programming interfaces (APIs) for programming the Integration Services object model.



There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Data Flow Task Task

ForcedExecutionVal	0
ForcedExecutionVal	Int32
ForceExecutionValu	False
Identification	
Description	Data Flow Task
ID	{D3A88A1F-6C83-44[illegible]
Name	Data Flow Task
Misc	
BLOBTempStorageI	
BufferTempStorage	

Name
Specifies the name of the object.

Figure 3.5 Microsoft SQL Server Integration Services

During this time I also got an opportunity to work as a user for a particular governing body website in uploading the relevant data. The situation is discussed in chapter 4 in detail which made me think of data migration where the DBA's expertise is needed. In such a case the focus is not on ETL but more on the fetch records, map onto destination DB and migrate.

The above listed DBA tool examples from 3.3.4.1 to 3.3.4.5 are among a wide variety of products available to be selected as per the need and demand of the situation. My consideration for them has solely been for the purpose of knowing the solutions available to the DBA for the administration task. But at the same time I have kept my focus on the data migration aspect related to DBA. **The tool discussed in 3.3.4.1 gives a good idea of data migration application**. There are other data migration tools also applicable at different levels or for different situations. They are Host-based file-level migration for static unstructured content, Host-based block-level migration for migrating one storage array to another, Network-based file-level migration, Network-based block-level migration for moving data online and Array-based block-level migration. After my study on such DBA tools and keeping data migration in view, I moved ahead for understanding the architecture of a few selected tools.

3.4 Comparison of any two DBA tools

In order to limit the scope of study and focus on the main aspect applicability of SOA to DBA tools, I have selected two tools based on the networking and DDB types. They are:-

3.4.1 Aqua Fold - Aqua Data Studio

Aqua Data Studio is a complete Integrated Development Environment (IDE) for DB query, administration and development. It is not aimed at monitoring but for enabling the DBA to manage manage the DBs. It supports all major DB vendors and works on all major operating systems. Aqua Data Studio scales with your ever growing datasets and can query and manipulate Big Data effortlessly. It is used for DB management by over 300,000 IT professionals in 98 countries, across Fortune 1000, public enterprise, science and technology sectors.

3.4.2 Redgate SQL Monitor

Redgate SQL Monitor is a web-based monitoring and alerting tool for SQL Servers. It displays real-time data about the current performance of all host machines, SQL Server instances, and DBs that you choose to monitor, and raises alerts when problems occur. SQL Monitor comprises three main components that need to be installed. They are Web Server, Base Monitor service, and Data Repository DB as shown in Figure 3.6.

The Web Server and Base Monitor service can be installed on the same computer, or on different computers on the network. The SQL Server DB for the Data Repository can be hosted on any SQL Server instance, but the Base Monitor computer needs access to this DB. Installation of the Web Server and

Base Monitor service on the same computer that's hosting the SQL Server instances to be monitored is not recommended.

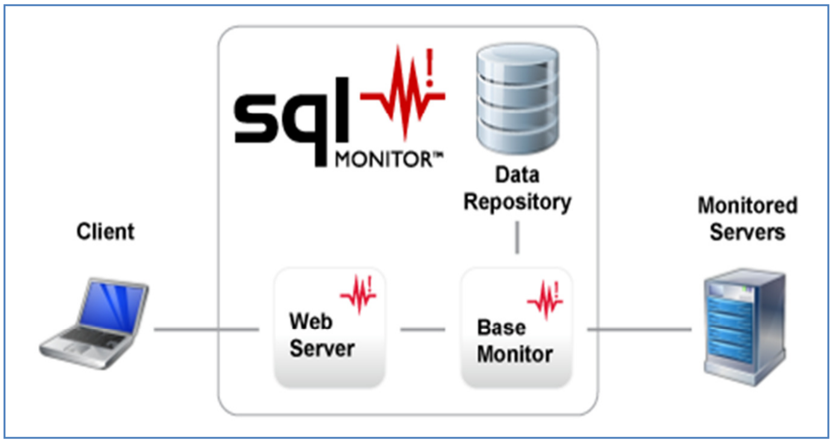


Figure 3.6 Redgate SQL Monitor Architecture

These tools have a number of functions and facilities for the benefit of the DBA. The selection of any of the tool depends on company requirements and business policies. I have shown a comparative of these tools considering their evolution, features and facilities and also their inclination towards open source development. The Table 3.1 is as follows :

Table 3.1 Comparison of Aqua Data Studio with Redgate SQL Monitor

Sr. No.	Parameter	Aqua Fold - Aqua Data Studio 14.0	Redgate SQL Monitor 3
1	Purpose	IDE for DB query, administration and development.	SQL Monitor is a web-based monitoring and alerting tool for SQL Servers.
2	Brief History	Version 6.5 in 2007 for Version Control Support for Subversion & CVS, Pivot Grid & Graphs, Grid Graphs, Grid Aggregate Functions, SQL [History] Archive, Results Compare Tool, Sybase & Oracle Debuggers to version 14.0 in 2013 for NoSQL DBs.	Version 1.0 in 2008 for SQL server monitoring to Version 4.0 for web-based monitoring and alerting for SQL Servers
3	DB support	DB2, MS SQL Server, SQL DB (Azure), MongoDB, MySQL, Netezza, Oracle, ParAccel, PostgreSQL, SQLite, Sybase,	SQL Server

		Teradata, Vertica, Apache Cassandra, Apache Derby, Apache Hive, Greenplum, Informix	
4	DDB Type	Heterogeneous (purpose is managing individual DB technicalities)	Homogeneous
5	Platform	Windows, Linux (JDK / JRE 1.6), Mac (Java 1.6)	Windows platform with .NET 3.5 SP1
6	Download	Respective 32 / 64 bit version	Installs three main components of SQL Monitor on a PC that is not a server itself
7	Network / Accessibility	Intranet	Internet. [SQL Monitor is a web application accessed using a web browser within your LAN.]
8	Open Source	$499 USD per user [AquaFold provides no-cost licensing to qualified individual OSS (open source software) developers].(as on 31/03/2014)	1 license $1,119. SQL Monitor is licensed and priced per server, with volume discounts available.(as on 31/03/2014)
9	Registration of DB Servers	Yes	Yes
10	Replication	Yes	Yes
11	Migration	Yes	No
12	Monitoring	Yes	Yes
13	Messaging	Yes	Yes
14	Email	Yes	Yes
15	User Roles	Yes	Yes
16	Security	Yes	Yes
17	Portability	No	Yes

The Table 3.1 gives a good idea as to the features and environment of the two tools. As per the study of the above DB administration and monitoring tools, we get a comparative of two existing DBA tools. The technical factors (9 to 17 of Table 3.1) in the comparison show that the DBA benefits immensely by way of monitoring the DB activity and taking prompt and necessary decisions for the smooth functioning of the system. The role and responsibility of the DBA is clear, precise and extremely important. ***This aspect directed me towards an architecture that is SOA based and would surely help in the development of more such functionality.***

The first step was in identifying the key components of the Intranet based DBA tool. The software for this tool had to be downloaded as per the platform and then installed on a machine to carry out further DBA activities. Normally, the user realizes that the architecture is client server based where information about various DB servers has to be registered and maintained. In order to identify the DB servers, the DB client needs to be present on the machine on which the tool is installed. The overall architecture can be summed up as shown in

Figure 3.7.

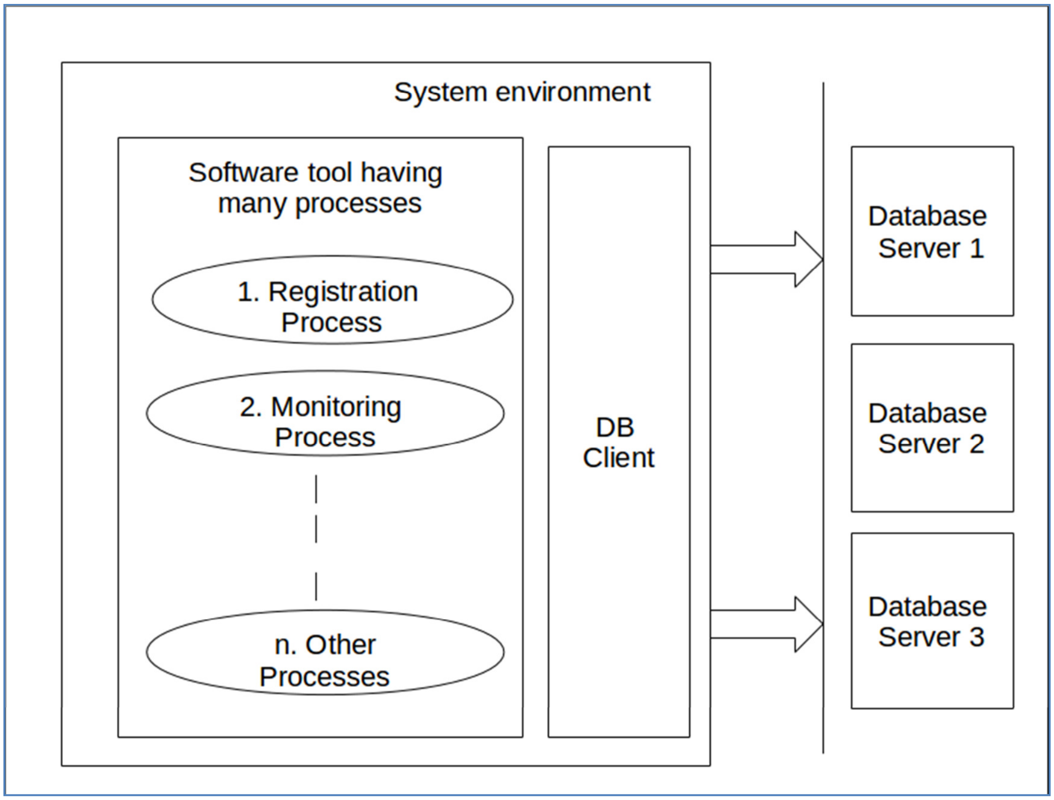


Figure 3.7 Two Tier Architecture based DBA Tool

The above architecture in

Figure 3.7 shows a few services that are of use to the DBA for the administration and monitoring activity of DDB servers. Similarly, for a web based architecture for DBA tool as shown in Figure 3.8, the user realizes the use of some service, Web Server and Data Repository. This tool is used to monitor the DB servers from any client machine that is accessible to the web server. It requires either a DB client or a xml based server group file to identify all the DB servers for further registration and administration.

In this architecture, the DBA needs to arrange for the three prerequisites mentioned in Figure 3.6 and then connect to the web server from any client in the company network. The base monitor service that is installed on the web server or other computer, needs to connect to the various SQL servers to be monitored and also connect to the data repository for storing data related to the SQL monitor. Finally, the client machine will display all the DBA related activities like monitoring.

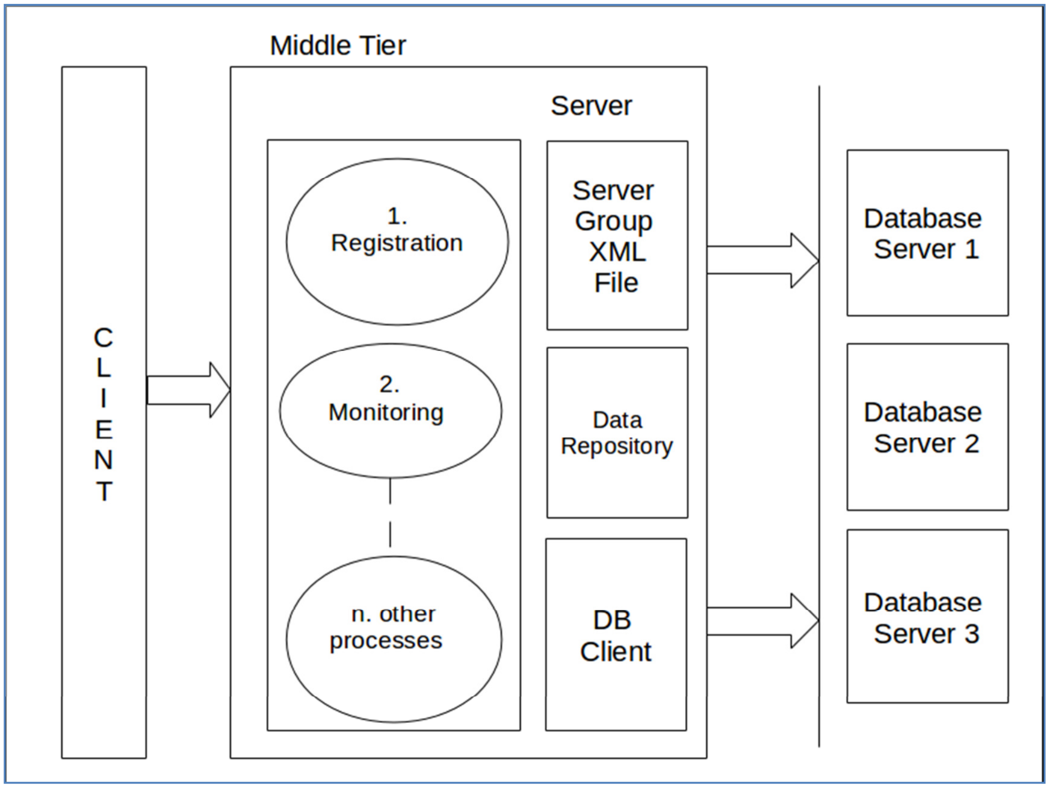


Figure 3.8 Three Tier Architecture based DBA Tool

Taking the reference from the architectures shown in Figure 3.7 and Figure 3.8*, I have worked upon the SOA based architecture for the planned (discussed in chapter 4) tool in order to migrate data of heterogeneous DDB* *under consideration.* The main requirements of the architecture are: 1) web server where services are deployed, 2) a DB or data repository where the data related to the tool and its activity can be stored and 3) the client from where the system can be accessed.

The web server and DB may be rented from a cloud space renting company so as to get the benefits of SOA totally. The benefit of scalability is also realized with this facility. *At the end of this chapter, I continued to concentrate more on the SOA based architecture and efforts in data migration. At around the same time, I got an opportunity to use a governing body software application for the student data upload facility. This really motivated me to think more and the further discussion is in the next chapter.*