Database Migration : An In Depth look!!

By

Anil Mahadev

anilm001@gmail.com

As most of you are aware of the fact that just like operating System migrations are taking place, databases are no different. We are at a time in the industry that needs data to be migrated from disparate systems of one platform to another. So what does one do? Migrate the Database. Migration is the most crucial and adventurous activity for a DBA, if you are a DBA, then you know exactly how one feels, when it comes to Database Migration.

One must be prepared for the kinds of pitfalls that they may face during the phases of migration. Mind you Database Migration is a highly skilled operation and must be done by people who have considerable Database Experience. Never ever compare database migration to any other type of migration, like OS backup concepts etc.
Database Migration is a skill in itself and comes by experience only.

For many of you who are new to Database migration, do not fear, start migrating small databases to large ones and experience is your best teacher :). Never Give up, keep trying!! and you will succeed.

So why an article on Database Migration?

The idea here is get the audience acquainted with the concept of the principles of migrating a database from one vendor to another.

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

Introduction:

The term ***"migration"*** is the process of moving from one location to another. The same applied for databases or operating systems, what have you.

Database Migration is serious business. One cannot afford to ignore the fact that, in our Database World, almost 30 -40 types of databases are available(both closed and open source ones). Therefore there will be a time, when organizations need to address their enterprise database needs, and provide their customers the database of choice.

There are enterprises today moving from proprietary databases to open source ones and some of them would like to move from one proprietary database to another. For example: Organization A uses SQL Server and would like to migrate to an IBM DB2 database. This can be done in a variety of ways.

It has been said that a Database Migration must be a well defined strategy. Just as Software, Databases too need to be taken care during the planning stages of any Development project.

It is therefore **crucial** that you understand the pro's and cons before implementing a Database Migration solution and how it will affect your enterprise.

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

We shall explore a Typical Migration Architecture as shown below in Figure 1.1

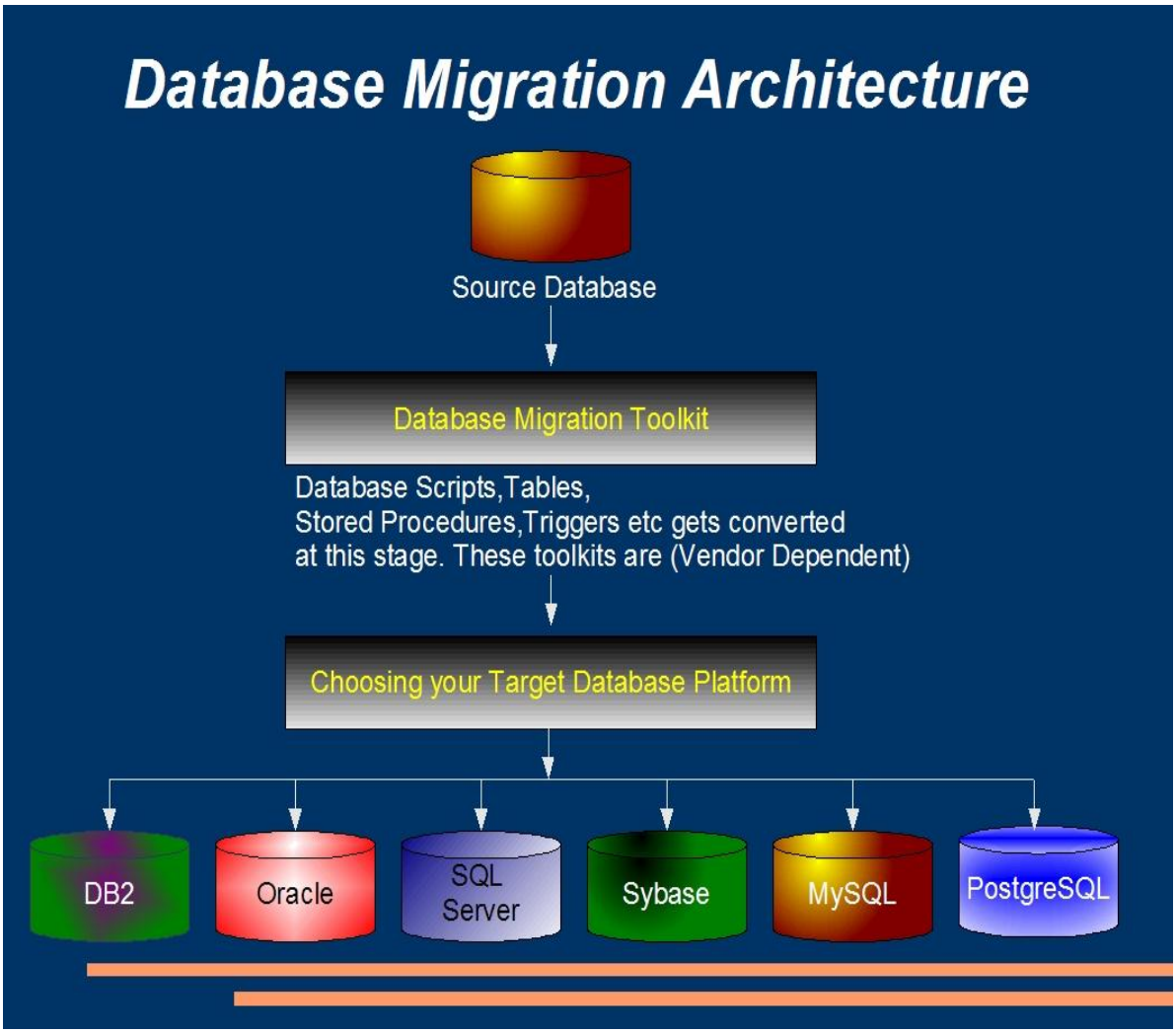


Figure 1.1

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

We shall discuss the following key points in this article:

- Key Benefits of a Migration.
- What do you need to get started ?
- Choosing your source and target databases for the migration.

  and

- Choosing the right tools for a Database Migration.

Key Benefits of a Migration:

1. Helps the enterprise choose the Database platform of choice.
2. Cost Savings from the previous database as compared to the target migration database.
3. And lastly , technological superiority of the migrated database.

It is important to ask yourself these questions?

1. When not to migrate?
2. When to migrate?

When not to migrate?

- It is important to realize that one has the choice not to migrate.
- If your current system is doing fine, then great :) , migration is not an option for you.

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

When to migrate?

- If your current database is not meeting the required performance levels you expected when you implemented it in your enterprise, or if high cost of maintenance and if the Database you are using is crunching your IT Budgets, then migrating to another database would be an option.

What you need to get started?

Well first of all you need a migration plan in place. Then you will need to decide on your source and target databases.

A migration plan would usually consist of the following items.

1. Database Architecture Maps of both the Source and Target Databases.
   The above include Logical and Physical Database Diagrams of the source and target databases.

2. A well designed Table of Contents of Database objects with their source database's data types,constraints and referential integrity keys and their proposed target database's data types,constraints and referential integrity keys.

3. Standards Document for Naming Conventions etc.

   and finally most important of all;

4. The source and target database in their consistent states. This is crucial to have in the case where, your users or customers are not going to be using that particular database any longer.

   (Note: It's assumed that your DBA has indeed done all the necessary steps to backup the source and target databases to be prepared for the migration)

With that being said, you will need to assemble a group of Database experts. A typical Migration Team would consist of a Database Architect, one or two DBAs from each platform and the CTO accompanied by CIO for Approving the final Migration Project.

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

Choosing the Source and Target Database platform:

Now that we have a Team, we need to choose the source and database platforms.
Any platform, I just don't mean the product itself, but the operating system as well.
We need to keep in mind the various constraints and benefits that will affect the migration process.

For example: If our target platform(both database and operating system) is DB2 and Linux, then we need to consider the performance benefits and return on investment with these two platforms.

For our article's purpose, we shall choose our source database as SQL Server and target database as DB2.

As most of you are aware that there will be fundamental differences between databases.

The most important thing to keep in mind are data-types, I repeat Data Types. If one is not careful, this could lead to disaster if the right data type mapping is not done.

To avoid the above hassles, develop a chart that will contain the source database data types on one side and the target database data types on the other.

Have a clear idea on what the implications would result in, for example: For **SQL Server**, if we have the **char** data type, one should also know that **DB2** also supports the **char** data type as well. The key here is not in the name of the data type , but the size of the data type :).

As the saying in our Database World goes **' Size does matter ! '.**

In the figure 1.2 below we can see, how a typical migration process occurs between SQL Server to IBM DB2.

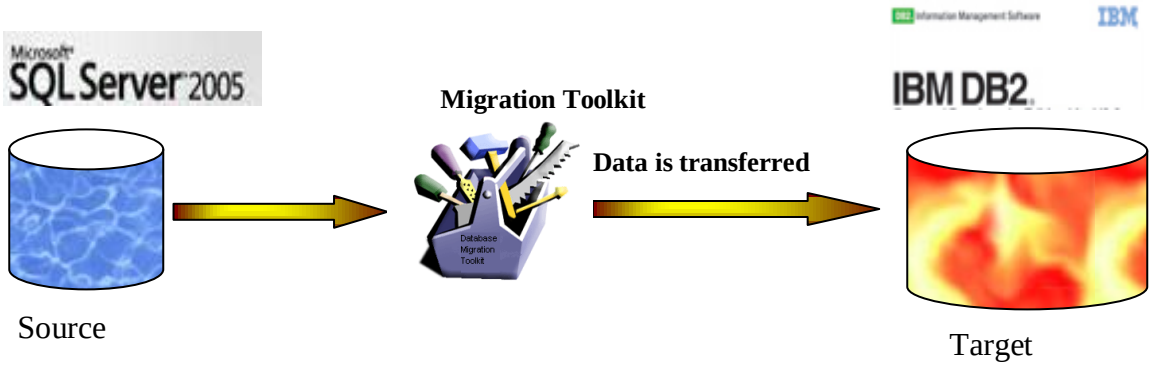


Figure 1.2

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

Choosing the right tools for a Database Migration.

We now move on to our final section of the article on what the various tools available for Database Migration. The fact of the matter is there are host of tools both vendor neutral and vendor based.

The choice is yours, as the saying goes, **"Customer is KING".**

Personally, as a DBA, I would usually recommend to go with Vendor based migration toolkits. For example to migrate from a SQL Server Database to DB2, one can use the IBM DB2 Migration Toolkit.

You can download a no charge copy from the following website.

http://www-128.ibm.com/developerworks/ondemand/migrate/db.html

Here are some of the vendor neutral Data Migration Tools available and one of my personal favorite list.

1. Swiss SQL: SQL Server to DB2
   http://www.swissql.com/products/sqlserver-to-db2/sql-server-to-db2.html

2. RealSoft Studio: SQL Porter SQL Server to DB2
   http://www.realsoftstudio.com/Standard_SqlporterForDB2.html

There are a host of other migration tools available, for the article's reference I have mentioned only two, which I personally felt appropriate.

Now moving on to the actual implementation Scenario.

For this purpose I shall be giving you all a high level overview of the Swiss SQL Tool for which the Migration from SQL Server to DB2 is possible.

To download a free 30 day trial, you can visit the following website

http://download1.adventnet.com/products/sqlserver2db2/55201272/AdventNetSwisSQLSQLServerToDB2_JRE.exe

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

Shown below is the screenshot of Swiss SQL: SQL Server to DB2 in Figure 1.3

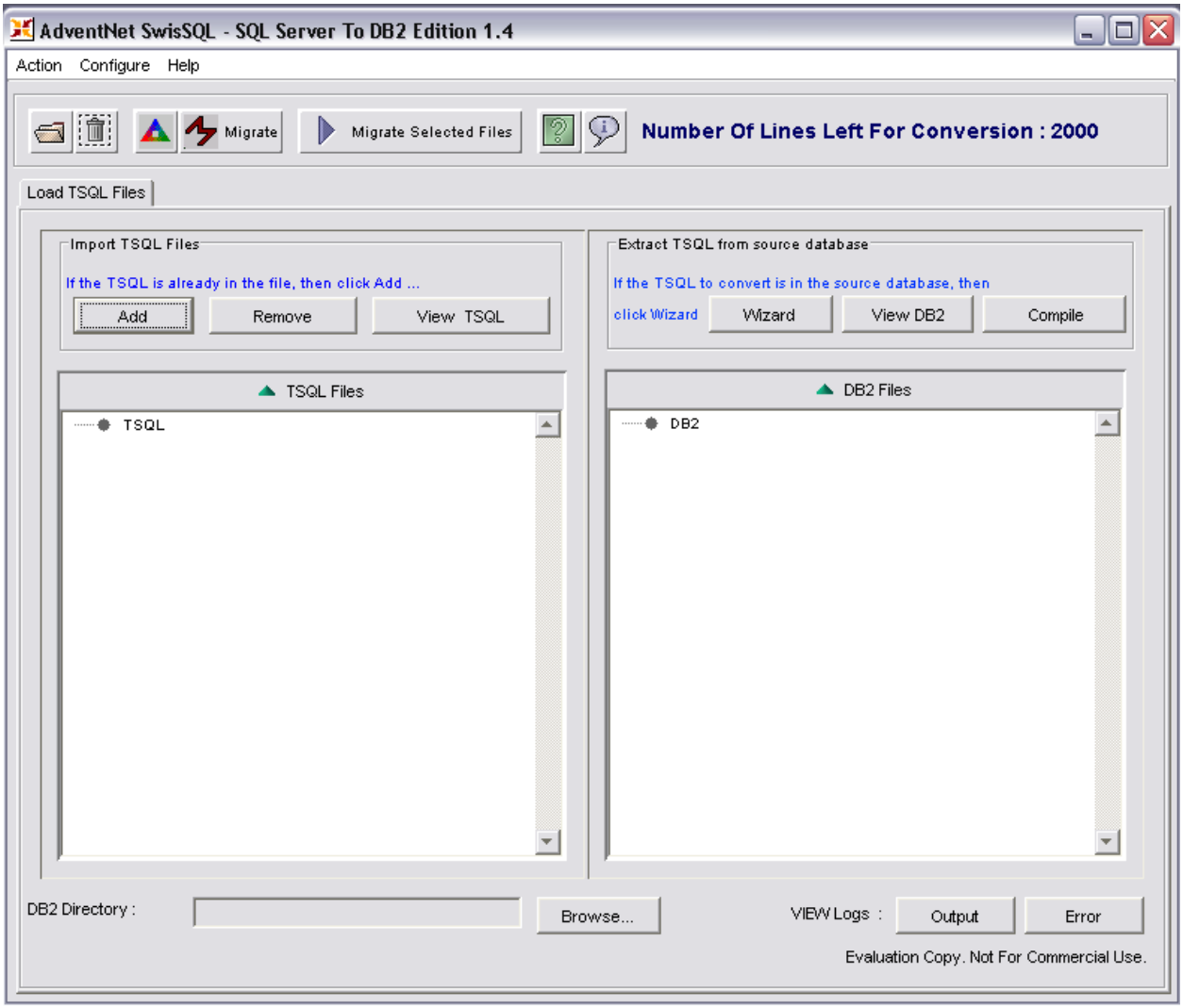


Figure 1.3

To start a migration project click on the button that says Quickly convert a database using the wizard.

You will be presented with a wizard like interface, that will explain on how to start a migration. Please note that that I will demonstrating creation of migration scripts from SQL

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

Server to DB2 and not Database to Database Migration. I have another tutorial in store for that very purpose.

We now take a look at on how to begin the migration process.
In the next screen, you can see that we shall connect using an ODBC connection. For this you will need to have an ODBC data source setup. The same is shown in figure 1.4

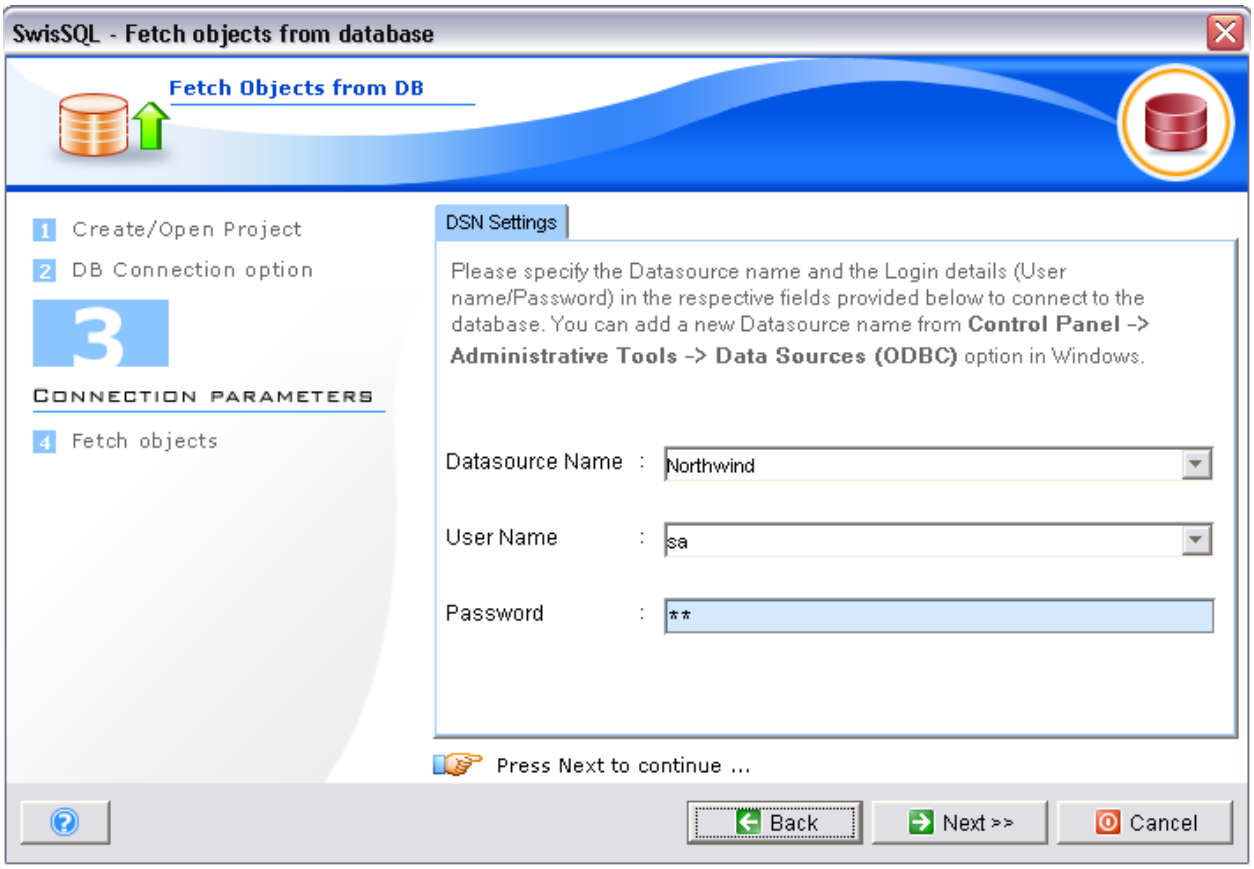
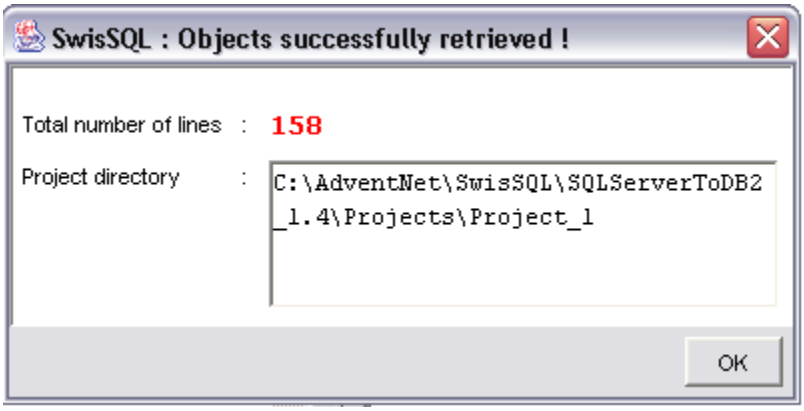


Figure 1.4

For the Data Source Name, please enter your DSN's name, for Username, enter your SQL Server Username and Password, your SQL Server Password.

After you have chosen this click next; You will be now taken to the object selection window, where one can specify which database objects to be migrated. For the purpose of this article I have chosen only Tables. This is shown in figure 1.5 below.

Figure 1.5



Once the objects are chosen, we are now left to complete the project. The final screen is shown below where it says it retrieved all the tables from the northwind database and has a total of 158 lines that needs to converted to DB2 SQL scripts.

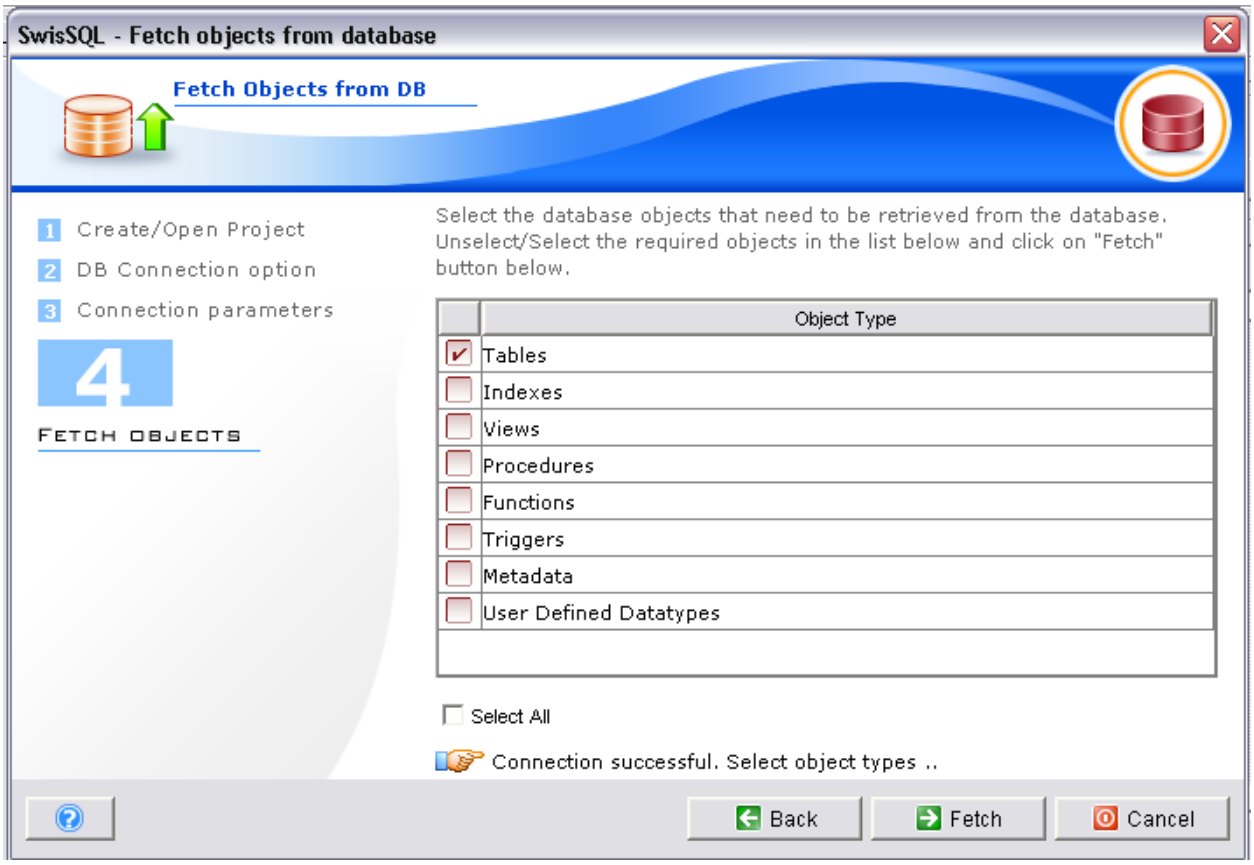
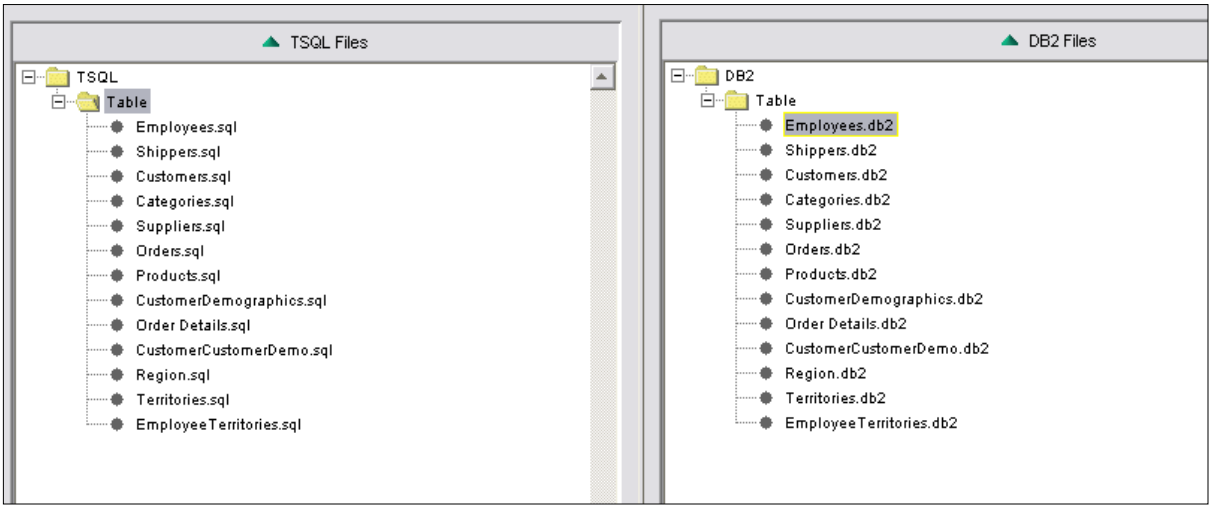


Figure 1.6

You now have the option to begin the migration. But before that you can check that indeed both the structures are in place as shown in the figure 1.7



Finally click on the button that says **Migrate.**
Next we shall migrate the Employees table of the Northwind Database to DB2.

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

In the table 1 show below is the SQL Server Script and DB2 Script

Table 1

SQL Server	***DB2***

CREATE TABLE [Employees] (
[EmployeeID] int IDENTITY(1,1) NOT NULL,
[LastName] nvarchar(20) NOT NULL,
[FirstName] nvarchar(10) NOT NULL,
[Title] nvarchar(30),
[TitleOfCourtesy] nvarchar(25),
[BirthDate] datetime,
[HireDate] datetime,
[Address] nvarchar(60),
[City] nvarchar(15),
[Region] nvarchar(15),
[PostalCode] nvarchar(10),
[Country] nvarchar(15),
[HomePhone] nvarchar(24),
[Extension] nvarchar(4),
[Photo] image,
[Notes] ntext,
[ReportsTo] int,
[PhotoPath] nvarchar(255)
, CONSTRAINT [PK_Employees] PRIMARY KEY ([EmployeeID])
, CONSTRAINT [FK_Employees_Employees] FOREIGN KEY([ReportsTo]) REFERENCES [Employees] ([EmployeeID])ON DELETE NO ACTION ON UPDATE NO ACTION
, CONSTRAINT [CK_Birthdate] CHECK([BirthDate] < getdate())
)
	CREATE TABLE Employees
	(EmployeeID int GENERATED BY DEFAULT AS IDENTITY(START WITH 1 INCREMENT BY 1) NOT NULL
	,
	  LastName VARCHAR (20) NOT NULL ,
	  FirstName VARCHAR (10) NOT NULL ,
	  Title VARCHAR (30) ,
	  TitleOfCourtesy VARCHAR (25) ,
	  BirthDate TIMESTAMP ,
	  HireDate TIMESTAMP ,
	  Address VARCHAR (60) ,
	  City VARCHAR (15) ,
	  Region VARCHAR (15) ,
	  PostalCode VARCHAR (10) ,
	  Country VARCHAR (15) ,
	  HomePhone VARCHAR (24) ,
	  Extension VARCHAR (4) ,
	  Photo BLOB (2G) NOT LOGGED
	,
	  Notes CLOB (1073741823) NOT LOGGED ,
	  ReportsTo int ,
	  PhotoPath VARCHAR (255) ,
	  CONSTRAINT PK_Employees PRIMARY KEY (EmployeeID) ,
	  CONSTRAINT FK_Employees_Employees FOREIGN KEY (ReportsTo) REFERENCES Employees (EmployeeID) ON UPDATE NO ACTION ON DELETE NO ACTION
	)

[Note: Before creating the database object in DB2, keep in mind the length of characters for constraints and column names.].

You your convenience, I have modified the DB2 Script and here it is show below in Table 2.

Table 2

```
CREATE TABLE ANILM.EMPLOYEES
(
 EMPLOYEEID INT  GENERATED BY DEFAULT AS IDENTITY(START WITH 1
INCREMENT BY 1) NOT NULL ,
 LASTNAME VARCHAR (20) NOT NULL ,
 FIRSTNAME VARCHAR (10) NOT NULL ,
 TITLE VARCHAR (30) ,
 TITLEOFCOURTESY VARCHAR (25) ,
 BIRTHDATE TIMESTAMP  ,
 HIREDATE TIMESTAMP  ,
 ADDRESS VARCHAR (60) ,
 CITY VARCHAR (15) ,
 REGION VARCHAR (15) ,
 POSTALCODE VARCHAR (10) ,
 COUNTRY VARCHAR (15) ,
 HOMEPHONE VARCHAR (24) ,
 EXTENSION VARCHAR (4) ,
 PHOTO BLOB (2G) NOT LOGGED ,
 NOTES CLOB (1073741823) NOT LOGGED ,
 REPORTSTO INT  ,
 PHOTOPATH VARCHAR (255) ,
 CONSTRAINT PK_EMP PRIMARY KEY (EMPLOYEEID) ,
 CONSTRAINT FK_EMP FOREIGN KEY (REPORTSTO) REFERENCES EMPLOYEES
(EMPLOYEEID) ON UPDATE NO ACTION ON DELETE NO ACTION
)

[Note: Replace ANILM with your Schema Name]
```

And now that we have our Employee Table Script ready, we need to run it in DB2.

You may choose to run it via the CLP,Command Window,Command Editor or the latest DB2 Workbench. For some of you who are not familiar with DB2 WorkBench, it is the latest offering Development Solution for Database Developers,DBAs who wish to create DB2 Objects, UDFs,Stored Procedures,Triggers and even the very best in XQuery Development.

DATABASE MIGRATION ARCHITECTURE By Anil Mahadev

DB2 Workbench replaces the DB2 Development Center. It is an Eclipse based IDE specifically catering to Database Developers who want to exploit DB2 9's pure XML capabilities.

So the first thing we need to do is to create a new Project just as in Eclipse, one would need to create a Database Development Project.

The benefit of using DB2 Workbench is that it can communicate with older DB2 servers and the ones running on AIX boxes as well :).

To run our script, we will establish a connection to the SAMPLE Database that one would create whenever we install DB2.

After you have created a new project , Navigate to the Connections Tab located on the bottom left corner of the IDE.
Expand it until you find your SAMPLE database connection information and now you should expand the Schema --> Tables --> Under that you will find the default tables. Now remember that DB2 has a table called **EMPLOYEE** and not **EMPLOYEES.** So remember to create this table as a new Schema object like x.Employee in your create table statement(replace x with your schema).

Now Right Click on the Tables Folder and Choose New --> SQL Editor. In the new SQL Editor copy and paste the above code, that i shall code in table 2.

Now run your DB2 Script and Voila, your new employees table that existed in the Northwind database of SQL Server is now in DB2 :).

Congratulations on making it this far !!!

In a forthcoming article I shall delve into the Intricacies of Logical Data Modeling of one database to another.

If you have any comments,suggestions,criticism, please do let me know and hope that this article gave you an insight into an Overview of Database Migration.

Database Migration is a topic in itself, so given the limited space, I would like to share more on the subject in the forthcoming issues of the Magazine :).

You may contact me AT anilm001@gmail.com

And please mention in the Subject Line Database Migration Article.