

**KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES
TECHNOLOGY**

ABODUNRIN S. DARE

**How to convert from Visual Fox Pro Database to
PostGreSQL**

6

Thesis of Information Technology Degree Program
Kemi 2012

PREFACE

The journey of a thousand miles they say starts with just a step and for every beginning, there must be an ending. This is obviously not flagging an end to my academic career but rather, a mark for the continuity in the realm of excellence in academic pursuit as this great citadel of learning as instilled in me.

To the success of this work, I hold a great measure of thanks to God who has constantly been my source of wisdom and strength in times of weakness.

Settlement is known as one of the greatest concerns of any student; it plays a major role in the measure of how any individual in any capacity turns out. I will therefore like to acknowledge the Kemi municipality for their endurance and also for creating a conducive environment to learn through my study period. Gratitude also extends to those who have collectively sown into me by taking out of their precious time the liberty to impart knowledge and to share their intellectual resources at every phase of the four years' study. The Students of KTUAS who acknowledged and accepted our existence in the neighbourhood, my thesis instructor, Mr Teppo Aalto whose immense guidance can never be quantified toward the successful completion of this work and most of all to Finland for giving me a first rate technical education in the field of Computing and Information Technology without demanding for penny to help realize the dream I have nurtured since childhood.

I say a big thank you to all.

Abodunrin Dare
Kemi, February 2012.

ABSTRACT

Kemi-Tornio University of Applied Sciences, Technology	
Degree Programme	Information Technology
Name	Abodunrin Dare
Title	How to convert from MS Visual FoxPro to PostGreSQL
Type of Study	Bachelor's Thesis
Date	16 February, 2012
Pages	37
Supervisor	Aalto Teppo, MSc

The fundamental idea behind computing is about processing and storing information which in an explicitly defined term is known as data manipulation. For several reasons, business to domestic, data transfer seem about the most important aspect of information processing such that a slight mistake in its handling could lead to a catastrophe for the intended user.

As the title depicts, conversion of data from a source to its destination database, this report carefully analyze both containers holding and receiving data and extends further to explain the procedures taken to ensure that the conversion is achieved by introducing a third database application which act as an intermediary to first accept content from the source then transfer it to the destination. Putting into consideration the complexity of this research work, clear explanations of all confusing concepts are given such as an introduction to all containers used, what a database is and how data migration is performed i.e. define and analyze source structure (data structure in the old system), define and analyze target structure (data structure in the new system), perform field mapping (mapping between the source and target), define the migration process (automated vs. manual).

To certify the correctness of information given in this report and its result, an adequate time was taken to discuss series of issues on each chapter presented with personnel's of greater understanding with which clarity was also given on logical concepts such as data structure, mapping and storage format. It would be observed at some latter instance the use of Linux commands during the transfer of data from the intermediary to the final destination, this was adequately reviewed by the supervisor who tested the outcome and also noted the preventive measures that should be considered to avoid data loss.

This thesis report serves as a descriptive guide to user(s) who might intend to work on a similar project or just to equip their knowledge of data conversion of any type. From the source to destination, all processes have been extensively highlighted and proper credit was given at the end to acknowledge writers of the books used and references to the internet resources.

Keywords: database, applications, data structure, linux commands, field mapping, migration process, automated, data management.

TABLE OF CONTENTS

PREFACE	i
ABSTRACT	ii
TABLE OF CONTENTS	iii
EXPLANATION OF CHARCTERS AND ABBREVIATIONS	iv
1. INTRODUCTION	1
Objective	1
Thesis structure	2
2. DATA MIGRATION	3
2.1. Migration Methodology	3
3. Microsoft VISUAL FOXPRO	5
3.1. Visual FoxPro database	5
3.2. Visual FoxPro tables	6
3.3. Visual FoxPro data types	7
4. POSTGRESQL	8
4.1. PostGreSQL database	9
4.2. PostGreSQL data types	9
4.3. PostGreSQL connectivity	10
5. MICROSOFT ACCESS	11
5.1. Microsoft Access database	12
5.2. Microsoft Access data type	12
5.3. Microsoft Access connectivity	14
6. CONVERSION FROM VFP TO POSTGRESQL	15
6.1. System requirements	15
6.1.1. Hardware and Software	16
6.1.2. Database Version and Utilities	16
6.1.3. Antivirus	16
6.2. Conversion from Visual FoxPro to Microsoft Access	16
6.2.1. Connection drivers	17
6.2.2. Conversion using ODBC connection	17
6.3. Conversion from Microsoft Access to PostGreSQL using ODBC	23
6.3.1. Install PostgreSQL ODBC driver	24
6.3.2. Create ODBC DSN for PostgreSQL	24
6.3.3. Export to PostgreSQL from Microsoft Access	26
7. CONCLUSIONS	27
8. REFERENCES	28

EXPLANATION OF CHARCTERS AND ABBREVIATIONS

ORDBMS	Object Relational Database Management Systems
RDBMS	Relational Database Management Systems
SQL	Structured Query Language
ODBC	Open Database Connectivity
ADO	ActiveX Data Objects
DSN	Data Source Name
OS	Operating System
VFP	Visual FoxPro

1. INTRODUCTION

Computers are used both in our domestic homes and at the industries for different reasons but its cogent purpose is to store and retrieve our data at any point on request. It is a surprising fact though that as much as the world has embraced the computer systems in this present technological era, most users still tend not to subject themselves to understanding the underlying data transfer or conversion concept which is acknowledged to be the basic parameter of a functioning system. Aside the cogent reason stated above, there are still many other reasons attached to data migration principle such as for better speed, performance, features and compatibility.

How to convert from Microsoft Visual FoxPro to PostgreSQL as the title reads presents a detailed data conversion process to readers from various professional backgrounds needing to expand their knowledge on the principle of data conversion. As a public document, readers are carried along by carefully introducing and analyzing each medium used at different conversion phases and then balances it with a practical work frame. Said differently, the theoretical aspect of this work comes first by introducing and explaining basic constituting concepts such as;

- What data is
- How information is stored in a database system i.e. file formats and containers
- How data is transferred i.e. making the holding containers open to read from
- How safe our data is when converted or transferred i.e. data migration

After that, a practical approach would be presented to display how the conversion is done realistically by using the three database applications aforementioned. The conversion model is thoroughly researched by consulting series of text books and resources on the internet.

Objective

The objective is to present an explicit guide on how to convert from a source database Microsoft Visual FoxPro to a defined PostgreSQL database application as its destination and also adequately clarify certain misconceptions relating to data transfer. To measure the degree of complexity saddled with, an extensive recall on prior knowledge accumulated over the years in the studied area of information technology such as; databases, computer architecture and programming skills are vastly deployed. It is intended to strengthen various intellectual minds in different area of specialization that might have some reasons to read this or to broaden their knowledge of data migration principle in general. Finally, this paper serves as a fundamental ground for advanced research work on this topic or similar for future use.

Thesis structure

Some of the chapters contained in this work are grouped together because of the similarities to what each entail. However, it is worth noting that each differs on subject matter and is a component of this thesis work. There are three different sections summarily in this research work and the first section mainly presents the introductory part. There are five chapters in the first section and each gives useful details on its subject. The first chapter gives an overall description of the thesis work, while the next defines the concept of data and how to migrate. Each of the remaining chapters introduces the three database applications used during the conversion processes and their data types. Chapter 6 is the only chapter in the second section and it explains theoretically the core conversion process. It is the main work and the technical aspect of this paper. It explains in detail the tools used, the processes involved and effective protection measure put in place to ascertain data safety. The last section is the round up phase which contains chapters seven and eight to conclude the thesis by presenting the conclusion and references respectively.

2. DATA MIGRATION

Data Migration for several reasons has been ascribed to human willingness for better services and this constant routine can be traced back as far as the inception of the computer system. A shift (relocation) in data center, or to balance workload, or to embrace a new technology i.e. the acquisition of a new operating system are just few of these noted reasons. Different areas of professionalism have their diverse definition for this concept but in a general term, data migration can be explained as the process of moving data from one device into another especially as a storage capacity grows and then redirecting the Input/output to the new system so that data can easily be read. /1/

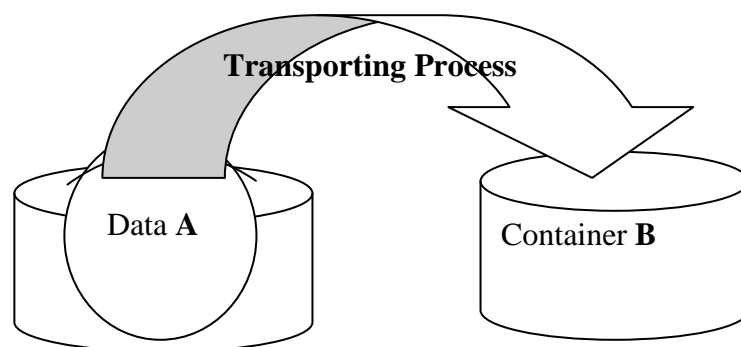


Fig. 1. Transportation Process of data from one container to another

Data transportation process is by far a complex subject that needs to be properly discussed and also the methodology used to ascertain that the right quality of data is delivered into our new system at the right time. However, data migration comes in two forms which are a classic migration and a consolidation migration. In classic, the data is moved into a completely new system from an old system. This old system is also known as the legacy system. A consolidated migration, on the other hand, is such that an upgrade is done by adding more data to a pre-existing one. Whatever migration form, a clear understanding of factors such as the data value and data volume must be considered. Those two factors help to pre-know how useful the new data is and how many datasets or records are needed to be moved into the new system. /2/ /3/

2.1. Migration Methodology

Migration methodology is simply a systematic guide line on the procedure observed to ensure that the data is transferred successfully from a repository container to a new destination just as explained above. Good planning is all it takes as it is observed that there is no difference in the method used for any type of migration. With a plan, it is easy to have a random view of all the phases, tasks, techniques, tools and an estimated time frame. So generally, the next phase after planning is to understand the migration environment i.e. information about the hardware and software. This is a very important step because it helps to clearly distinguish the source from the destination and also helps to understand the data to be transported. With those known, mapping and designing comes after. These phases

help to understand the mapping layouts concept and to choose from the two known layout models. For instance, there is a one-to-one layout and there is a relay layout. In one to one layout model, both the source and destination containers have the same storage layout. In relay layout model, both have different storage layouts. The diagram below presents a flow chart by breaking the method into phases for a clearer understanding. /1/

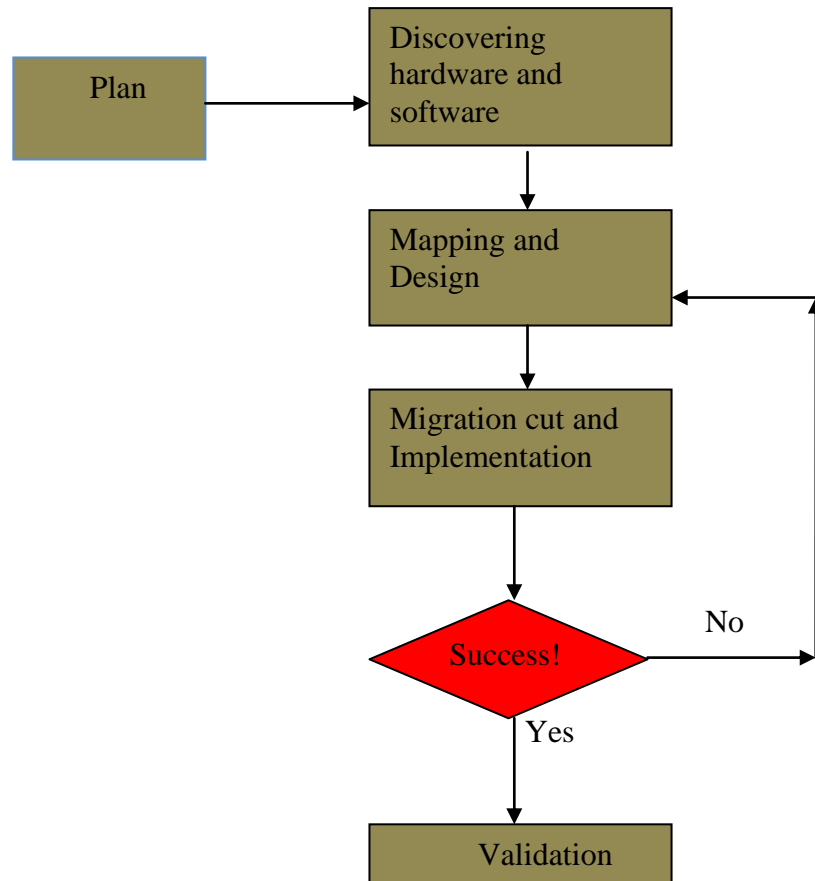


Fig. 2. Migration methodology

3. MICROSOFT VISUAL FOXPRO

Visual FoxPro (henceforth VFP) is a relational database management system and a programming language. As a database management system, it means that the VFP is a computer installed application used to control the creation, maintenance and use of database. With its capability as a programming language, VFP does not need additional external programming environment i.e. plug-ins to write its own files or to build other wide applications such as a web or any enterprise application. One of its uniqueness is that it has its own database engine which is dynamic enough to let it interact with not only itself but also extends support to the standard SQL query language creating an ease to data manipulation which by meaning, allows VFP to interact with SQL server, MYSQL, Oracle and any other Ole-DB accessible data source. /4/ /5/

Since the merger which made Microsoft the rightful owner, from the mid of 1992 till this present day of 2012, newer versions of the VFP software has been released to add more functionalities to the product. As newer versions are released starting from VFP 3.0 to the latest and last commercial release VFP 9.0, the platform support has been reduced drastically from different operating systems such as Mac, Dos, windows and UNIX to only windows. /4/

Table 1 below gives the list of a few windows operating system and the VFP versions supported. The numbers highlighted in red marks the various Visual FoxPro versions while the windows operating systems maintain the default black color. Yes is used to denote the compatibility while a null space means no.

Table 1. VFP versions and compatible Windows Operating System

VFP	3.0	5.0	6.0	7.0	8.0	9.0
WINDOWS						
3.1	Yes					
NT 4.0	Yes	Yes	Yes	Yes		
98	Yes	Yes	Yes	Yes		
SERVER 2003			Yes	Yes	Yes	Yes
7	Yes		Yes	Yes	Yes	Yes

3.1. Visual FoxPro database

Just like any other database systems, Visual FoxPro database provides a rich set of functionalities in its working environment which allows user to create and effectively

organize tables and views. VFP is a relational database system and therefore works by creating a relationship between all its tables in a particular database and also presents us with varieties of additional benefits to use. For clarity, a .dbc is automatically created by VFP each time we create a database and this .dbc (database connector descriptor file) is used to store not the real data but the file path which is the exact location of the files storing the physical items in the database. /7/

There are many file extensions and their associated file types in the VFP database system but in this paper concern is only on the .dbc, .fpt, .cdx and .dbf. Table 2 below shows a few of the extension list and file types. /8/

Table 2. VFP Files and Extension types

Extension	File type
.dbc	Database
.dbf	Table
.dcx	Database index
.fmt	Format file
.dct	Database memo
.ocx	OLE control
.fpt	Table memo

As mentioned earlier, VFP provides an unlimited set of rich functionalities and one of the newly added functions is discussed briefly below.

- **Data dictionary features**

This feature offers the possibility to specify items important to the database such as creating a long name for tables or fields, default values in the fields, default control class triggers and many more. A notable usefulness of this feature is that it helps to reduce the amount of codes written when attempting to modify or design a database. /7/ /8/

3.2. Visual FoxPro tables

When designing a database, it is required to specify the items it contains. One of those items is a table tray which uses a row and column format to store and deeply define an item. Tables in VFP database systems are saved as a .dbf. They usually exist in two forms; as a free table or a database table and as their names obviously indicate; a database table is a table linked to a database while a free table is not. A link is used to connect table files to a particular database. In a database table, the information for the link is stored in the database which uses a .dbc file extension format. This link can be a forward or backward link depending on where the information is stored. If the link information is stored in a database file then it is a forward link. It is considered backward if the link information is stored in a table file. This is just used to identify the location of the link information. /7/ /9/

3.3. Visual FoxPro data types

Databases are mainly concerned about data and how to store, manipulate and retrieve data when needed. Data varies in sizes for example; a bigger memory space is allocated to store a currency data when compared to the storage space needed for a character data type. Therefore, a database can only work efficiently if the right data types are assigned. In VFP, such is the case. Every data has its appropriate type assigned to it and this is done according to its size. Unlike the static data type assignment to variables and arrays, it is possible to specify data type for each field a table has. Visual FoxPro has seven general data types and a list of many data types for fields in a table. /10/

Table 3. VFP Data types

Data Type	Size	Range
Character	1 byte per character	Any characters
Currency	8 bytes	
Date	8 bytes	
DateTime	8 bytes	
Logical	1 bytes	True or False
Numeric	8 bytes in memory, 1-20 bytes in table	
Variant	Contains any of the VFP data types. So it equals whatever is stored in it.	Depends on the data type

4. POSTGRESQL

PostgreSQL, or Postgres as popularly known, is an object relational database management system (ORDBMS). It means that PostgreSQL usually interface by storing data as objects and the system converts that object information into data tables i.e. rows and columns when requested to use. This database management system is developed and maintained by the global community of developers because it is an open source and this gives the capability to run and work on various computing vendors unlike the other two database management systems also used in this paper. PostgreSQL is compatible with many operating systems and never stops growing because of its flexible licensing structure that allows any user to modify or add to its development. /11/

PostgreSQL is still a bit confusing to very many as it is perceived to be always the same as SQL. For clarity, PostgreSQL only supports a very large part of the standard SQL and this enriches its capability to serve us better by providing extended modern features which as a reason as compounded its popularity amongst other database systems. As the world's most advanced open source database server, it never stops growing as newer versions are released almost yearly and the beauty of this system is that it works on most of the available computer system architecture. The table below gives a clue to the major releases and years. /12/ /13/

Table 4. PostgreSQL major releases and years

Release	First release	Latest minor version	Latest release	Additions
0.01	1995-05-01	0.03	1995-07-21	Initial release as Postgres95
1.0	1995-09-05	1.09	1996-11-04	Changed copyright to a more liberal license
6.0	1997-01-29	—		Name change from Postgres95 to PostgreSQL, unique indexes, pg_dumpall utility, ident authentication.
6.1	1997-06-08	6.1.1	1997-07-22	Multi-column indexes, sequences, money data type, GEQO (Genetic Query Optimizer).
6.2	1997-10-02	6.2.1	1997-10-17	JDBC interface, triggers, server programming interface, constraints.
6.3	1998-03-01	6.3.2	1998-04-07	SQL92 subselect capability, PL/pgTCL
6.4	1998-10-30	6.4.2	1998-12-20	VIEWS and RULEs, PL/pgSQL
6.5	1999-06-09	6.5.3	1999-10-13	MVCC, temporary tables, more SQL statement support (CASE, INTERSECT, and EXCEPT)
7.0	2000-05-08	7.0.3	2000-11-11	Foreign keys, SQL92 syntax for joins
7.1	2001-04-13	7.1.3	2001-08-15	Write-ahead Log, Outer joins
7.2	2002-02-04	7.2.8	2005-05-09	PL/Python, OIDs no longer required, internationalization of messages
7.3	2002-11-27	7.3.21	2008-01-07	Schema, Internationalization
7.4	2003-11-17	7.4.30	2010-10-04	Optimization all-round
8.0	2005-01-19	8.0.26	2010-10-04	Native server on Microsoft Windows, savepoints, tablespaces, exception handling in functions, point-in-time recovery
8.1	2005-11-08	8.1.23	2010-12-16	Performance optimization, two-phase commit, table partitioning, index bitmap scan, shared row locking, roles
8.2	2006-12-05	8.2.23	2011-09-26	Performance optimization, online index builds, advisory locks, warm standby
8.3	2008-02-04	8.3.18	2012-02-27	Full text search, SQL/XML, ENUM types, UUID types
8.4	2009-07-01	8.4.11	2012-02-27	Windowing functions, default and variadic parameters for functions, column-level permissions, parallel database restore, per-database collation, common table expressions and recursive queries
9.0	2010-09-20	9.0.7	2012-02-27	Built-in binary streaming replication, Hot standby, 64-bit Windows, per-column triggers and conditional trigger execution, exclusion constraints, anonymous code blocks, named parameters, password rules
9.1	2011-09-12	9.1.3	2012-02-27	Synchronous replication, per-column collations, unlogged tables, K-nearest-neighbor indexing, serializable snapshot isolation, writeable common table expressions, SE-Linux integration, extensions, SQL/MED attached tables (Foreign Data Wrappers), triggers on views

4.1. PostgreSQL database

PostgreSQL offers a variety of features to ensure proper management of data, its retrieval and reliability as well as given standard security features. However, this database system uses a client server mode of interaction which makes it a bit complex to create interaction between the server and the client application. It is an object relational database management system therefore PostgreSQL does not use a single file schema to store its data. To explain this, a data in PostgreSQL typically contains a couple of SQL objects such as function, operator class index and operators. These three objects are packed together in a single jacket which is called an extension. These objects can be loaded into the database using the CREATE EXTENSION command as long as there is a script file, extension control file and a share library file. /14/

Data is stored in the PGDATA of a PostgreSQL database. This PGDATA contains lists of sub-directories which are used to store the location of databases' files. The table below lists known subdirectories and control files that are stored in the PGDATA. /14/ /15/

Table 5. PGDATA subdirectories and files /15/

Item	Description
PG_VERSION	A file containing the major version number of PostgreSQL
base	Subdirectory containing per-database subdirectories
global	Subdirectory containing cluster-wide tables, such as pg_database
pg_clog	Subdirectory containing transaction commit status data
pg_multixact	Subdirectory containing multitransaction status data (used for shared row locks)
pg_subtrans	Subdirectory containing subtransaction status data
pg_tblspc	Subdirectory containing symbolic links to tablespaces
pg_twophase	Subdirectory containing state files for prepared transactions
pg_xlog	Subdirectory containing WAL (Write Ahead Log) files
postmaster.opts	A file recording the command-line options the postmaster was last started with
postmaster.pid	A lock file recording the current postmaster PID and shared memory segment ID (not present after postmaster shutdown)

4.2. PostgreSQL data types

PostgreSQL has an extensive set of data types. By using the CREATE TYPE command, a user can create new data types aside the native ones. This gives the luxury to build a robust database and design more complex database applications. However, it is observed that the POSTGRESQL and SQL share some similarities in types i.e. the data type of one corresponds to the other as can be seen in table 6 below. /16/

Table 6. PostgreSQL data types /16/

Postgres Type	SQL92 or SQL3 Type	Description
bool	boolean	logical boolean (true/false)
box		rectangular box in 2D plane
char(n)	character(n)	fixed-length character string
cidr		IP version 4 network or host address
circle		circle in 2D plane
date	date	calendar date without time of day
decimal	decimal(p,s)	exact numeric for $p \leq 9, s = 0$
float4	float(p), $p < 7$	floating-point number with precision p
float8	float(p), $7 \leq p < 16$	floating-point number with precision p
inet		IP version 4 network or host address
int2	smallint	signed two-byte integer
int4	int, integer	signed 4-byte integer
int8		signed 8-byte integer
interval	interval	general-use time span
line		infinite line in 2D plane
lseg		line segment in 2D plane
money	decimal(9,2)	US-style currency
numeric	numeric(p,s)	exact numeric for $p \leq 9, s = 0$
path		open and closed geometric path in 2D plane
point		geometric point in 2D plane
polygon		closed geometric path in 2D plane
serial		unique id for indexing and cross-reference
time	time	time of day
timetz	time with time zone	time of day, including time zone
timestamp	timestamp with time zone	date/time
varchar(n)	character varying(n)	variable-length character string

4.3. PostgreSQL connectivity

PostgreSQL runs on most of the operating systems known today and can share its resources with other database management systems running on same or different platform. New versions of this relational database system come with improved functionalities thereby making transferring data by importing and exporting easy. There are many ways to do this, i.e. manually or with the use of already designed applications. There is also the possibility to use interface oriented connection approach such as ODBC, phppgadmin and many more. The application user connecting to PostgreSQL determines which connection method is best used. /15/

5. MICROSOFT ACCESS

Microsoft access is a relational database management system that has gone through different releases by Microsoft since inception in 1992. These modifications are simply to make the act of data management easy to reference stored information, report and analyze. It is one of the applications that come with the Microsoft office suite, although Microsoft Access is not available in all the suite versions. Most of the other database management systems work on more than one platform but Access only runs on windows. Nevertheless, because of its portability and easy open merger with other databases that also support the OLE DB connectivity, software developers find access as a very useful data management tool to interface during data migration between any two database systems or more. /17/ /18/ /19/

Microsoft access is best known to be a computer based database application but not strongly considered the best for internet based applications which are known to be data intensive and much more complex. Put differently, Microsoft Access can also be used to organize or manage enterprise data but some other database systems are considered stronger when considering the management of data intensive network. Other database management development companies never stopped releasing newer versions of their database application Microsoft also has many versions of access database systems released. The versions are listed below in table 7 with their respective supported OS and year of release. /17/ /18/ /20/ /21/

Table 7. Microsoft Access releases and years

Version	Year of Release	Supported OS
Access 1.1	1992	Windows 3.0
Access 2.0	1993	Windows 3.1x
Access for windows 95	1995	Windows 95
Access 97	1997	Windows 9x, NT 3.51/4.0
Access 2000	1999	Windows 9x, NT 4.0, 2000, XP
Access 2002	2001	Windows 98, Me, XP, 2000
Access 2003	2003	Windows 2000, XP, XP Professional x64, Vista
Access 2007	2007	Windows XP SP2, XP Professional x64, Vista
Access 2010	2010	Windows XP SP3, Vista and 7

5.1. Microsoft Access database

Microsoft access database handles its data just as every other relational database system does. It stores information, creates relationship between them and makes it available to users when requested. One of its seven general components is the table which the database system uses to store data. This table feature is secured in Microsoft Access and each field can be configured desirably to accept a precise type of information to be stored in it. Access compatibility with the SQL standards gives the possibility to manipulate data in the access table using SQL statements and also shares data with other applications or database systems by importing or exporting. /19//20//22/

Unlike some of the other relational database systems that use client-server software model, Microsoft Access is a file-server database system. This means that the database is a window file and it is stored in a location in the computer system which is connected to a network for data reference when needed. The file extension type associated with the database system depends on the version of Microsoft Access available. In table 8 below, series of file extensions for different version of the Microsoft Access is given. /17/ /21/ /23/

Table 8. Microsoft Access files extensions

File Format	Extension
Access Project	.adp
Access Blank Project Template	.adn
Access Database 2007	.accdb
Access Database Runtime 2007	.accdr
Access Database Template 2007	.accdt
Access Database (2003 and earlier)	.mdb
Access Database, used for ad dins (2, 95, 97), previously used for workgroup (2)	.mda
Access Workgroup, database for user-level security.	.mdw
Access (SQL Server) detached database (2000)	.mdf
Protected Access Database, with compiled VBA (2003 and earlier)	.mde
Protected Access Database, with compiled VBA (2007)	.accde
Windows Shortcut: Access Macro	.mam
Windows Shortcut: Access Query	.maq
Windows Shortcut: Access Report	.mar
Windows Shortcut: Access Table	.mat
Windows Shortcut: Access Form	.maf

5.2. Microsoft Access data type

Microsoft Access, data is stored in tables and these tables are made up of fields. Fields are created using two parameters; field name and field data type. These two proprieties define

the field's behavior but the more important of the two is the data type. Microsoft Access has ten different types of data that are shown in table 9 below. /24/ /25/

Table 9. Microsoft Access data types

Data Type	Usefulness	Size
Text	Short, alphanumeric values, such as a last name or a street address.	Up to 255 characters. Microsoft Access only stores the characters entered in a field; it does not store space characters for unused positions in a Text field.
Memo	Lengthy text and numbers, such as notes or descriptions.	Up to 64,000 characters
OLE Object	Objects (such as Microsoft Word documents, Microsoft Excel spreadsheets, pictures, sounds, or other binary data), created in other programs using the OLE protocol, that can be linked to or embedded in a Microsoft Access table	Up to 1 gigabytes
Number	Numeric data to be used for mathematical calculations, except calculations involving money (use Currency type).	1,2,4,8 bytes. 16 bytes for replication of ID (GUID) only
Date/Time	Date and Time	8 Bytes
Hyperlink	Field that will store hyperlinks. A hyperlink can be a UNC path or a URL	Up to 64,000 characters
Currency	Currency values. Use the Currency data type to prevent rounding off during calculations. 4 digits	8 Bytes
Auto Number	Unique sequential (incrementing by 1) or random numbers automatically inserted when a record is added.	4 bytes. 16 bytes for replication ID (GUID) only
Yes/No	Boolean values	1 bit
Lookup Wizard	A field that allows you to choose a value from another table or from a list of values using a combo box.	4 bytes

The numeric data type also has different defining specification. Numbers could be whole, decimal, long and many more. For this reason, table 10 also shows those further simplifications for a numeric data type.

Table 10. Microsoft Access numeric data types

Type	Description	Size
Byte	Stores 0 to 225, no fractions	1 byte
Integer	Stores -32,768 to 32,767, no fractions	2 bytes
Long Integer	(Default) Stores number from -2,147,483,648 to 2,147,483,647 no fractions.	4 bytes
Decimal	Stores numbers from $-10^{38} - 1$ through $10^{38} - 1$ (.adp) Stores numbers from $-10^{28} - 1$ through $10^{28} - 1$ (.mdb)	12 bytes
Single	Stores numbers from $-3.402823E38$ to $-1.401298E-45$ for negative values and from $1.401298E-45$ to $3.402823E38$ for positive values.	4 bytes
Double	Stores numbers from $-1.79769313486231E308$ to $-4.94065645841247E-324$ for negative values and from $1.79769313486231E308$ to $4.94065645841247E-324$ for positive values.	8 bytes
Replication ID	Globally Unique Identifier	16 bytes

5.3. Microsoft Access connectivity

One of the characteristics of a good database system is to have a flexible data sharing attribute with other applications that might be running on same or a different platform i.e. on same or different computer systems. Microsoft Access has this connectivity enhancement which gives the opportunity to import and export data to and from its database. One of the many possible approaches is to use the Open Database Connectivity (ODBC) driver although there is also ADO connection possibility. There are many applications developed already to help automatically link, read and write data from Microsoft Access and other database systems but manually, by using the ODBC driver with SQL statements makes migration of tables or database possible. /26/

6. CONVERSION FROM VFP TO POSTGRESQL

This chapter deals with the theoretical explanation of the conversion process from a commercial Visual FoxPro database system to an open source database, PostGreSQL. Microsoft Systems Inc. did not create the option to save a .dbc file to a PostgreSQL database format making the EXPORT / COPY TO command in visual FoxPro incapable. That presents us with a problem of how to convert data i.e. set of tables from the aforementioned source database system to the destination which is the PostgreSQL.

Problem: How to convert from Visual FoxPro to POSTGRESQL

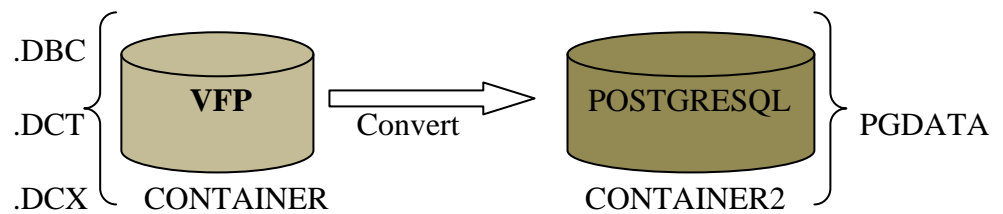


Fig. 3. Source and destination conversion

An attempt to solve the problem forms the basis of this research documentation where the solution takes the form given below.

Solution: Convert from VFP to MS Access to POSTGRESQL

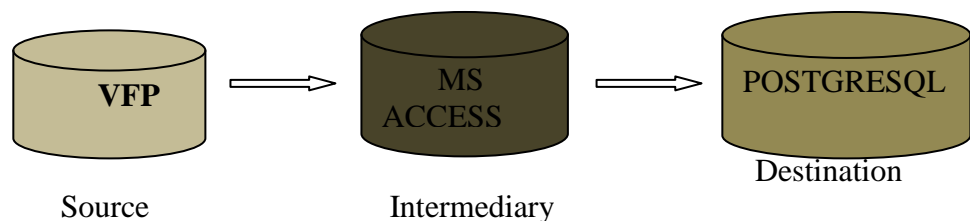


Fig. 4. Conversions break down

6.1. System requirements

Like any other computer related project, migrating data from one database system to another requires proper planning which helps to put the right set of equipment in place to

ensure the data safety during migration processes. As previously stated in a chapter that dealt with data migration methodology, a concrete plan points out in clarity certain facts such as what the problem is, how to solve it and what is needed.

6.1.1. Hardware and Software

The first thing to consider is the computer system in place. For a migration of this magnitude which involves the use of three different database applications. The system should be modern and strong enough to support each application's architecture so that these applications can discharge in response to time duties assigned them. Another important aspect to consider is the Operating System that runs on this machine and its compatibility with this three database applications i.e. Visual FoxPro, Microsoft Access and PostgreSQL. /27/

In this project, the system specification and running operating system information is given below.

Computer Model: Dell Latitude D630

Operating System: Windows 7 professional, service pack 1

Processor: 2 Duo CPU @ 2.2GHZ, 2.2 GHZ

RAM: 2.00 GB

System Architecture Type: 32 bit Operating System

6.1.2. Database Version and Utilities

There are many versions available for each of the database systems intended to use, hence choosing the right version is necessary for compatibility, speed and high performance of the operating system that runs in the computer during conversion. Visual FoxPro 9.0, Microsoft Access 2010 and PostgreSQL 9.1.3 are the selected versions for this project and these three are compatible with windows 7 with several rich features.

6.1.3. Antivirus

To make sure the system is not vulnerable to attacks, i.e. viruses while searching for materials on the internet, a good antivirus tool should be installed to minimize the possibility of an attack and to make sure the files are not corrupted which can lead to data loss or system crash. The antivirus tool used on my system is the Microsoft Security Essentials. This antivirus software performs real time and on demand a virus check.

6.2. Conversion from Visual FoxPro to Microsoft Access

As indicated in the solution diagram given above, there are two phases in the conversion process. It should be clear enough at this point that the words conversion, migration and transfer are used interchangeably. In the first phase, migration is done from the source database to the intermediary which is Visual FoxPro and Microsoft Access respectively.

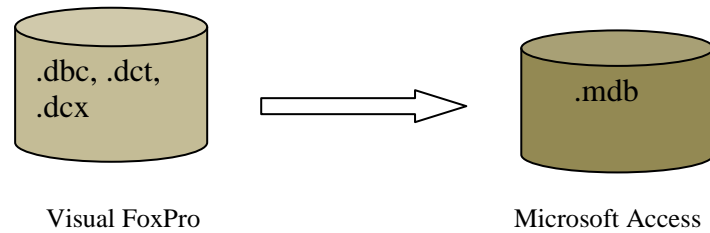


Fig. 5. First phase of conversion

It should be noted before commencing on the explanation of the proposed solution that there is a ready designed database in Visual FoxPro, so there will not be any explanation on how to create a Visual FoxPro database since the major concern here is to convert data from one application to another.

Tables are what moves during data transfer from a Visual FoxPro database to another. This table is structured to have vertical and horizontal columns where data information and values such as a unique key index and many more are stored. Before moving tables, certain drivers would have to be downloaded to enable reading and writing from the databases to create connectivity.

6.2.1. Connection drivers

There are many versions for each product of database system chosen and since newer versions come with better features, there is better utility to enhance connectivity in most of the recent model available. Here, we have VFP 9.0, Ms Access 2010 and PostgreSQL 9.1.3. All of those are recent versions yet there is no direct link between the source and destination system. The only way to adjust this is by having a connecting driver that is independent of the Operating system or the computer system. For the connection driver, ODBC and OLE DB drivers are installed to link and transfer data between those aforementioned database systems although one driver is used at a time.

6.2.2. Conversion using ODBC connection

Both applications run on the same computer system, i.e. Visual FoxPro 9.0 and Microsoft Access 2010 runs on windows 7 operating system. ODBC which stands for Open Database Connectivity has an Interactive Interface which is known as the "ODBC Client" and a driver that interpret commands sent by the user to the database management systems. Since Microsoft Access is the intermediary database management application that will be used to read and transport data from the VFP, it is therefore required to install the VFP ODBC driver and configure it to work. /32/

The first step is to create an ODBC connection and configure it. Different operating systems have a different way of navigating to the utilities in their control panel for this purpose. On windows 7, click the start menu icon, go to the control panel, click on the system and security option, click on the administrative tools option and on the list is a "Data source (ODBC)". A window as shown below appears after clicking.

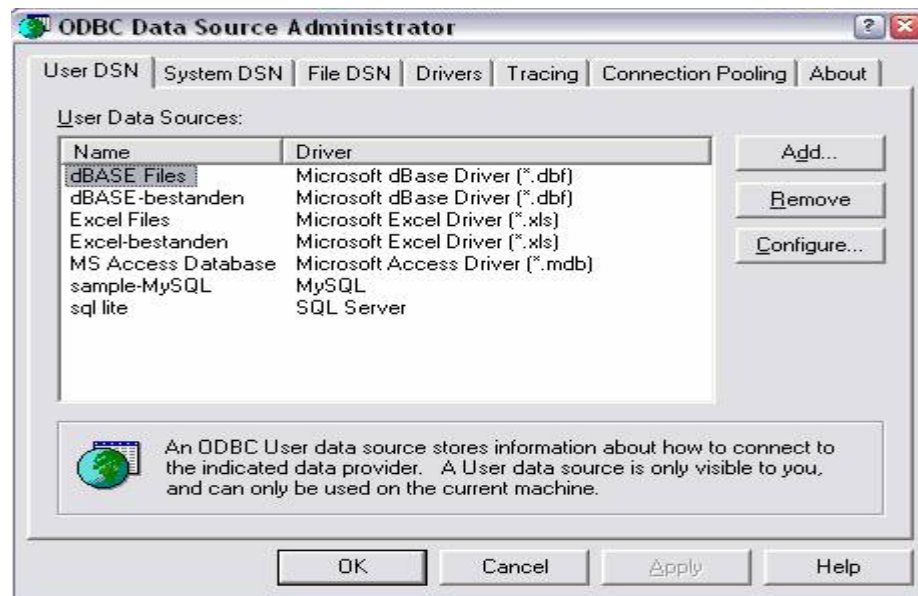


Fig. 6. ODBC Administrator interface

The “ADD” button allows the user to select the database driver for connecting. So after clicking the add button, a window as shown below in figure 7 appears.

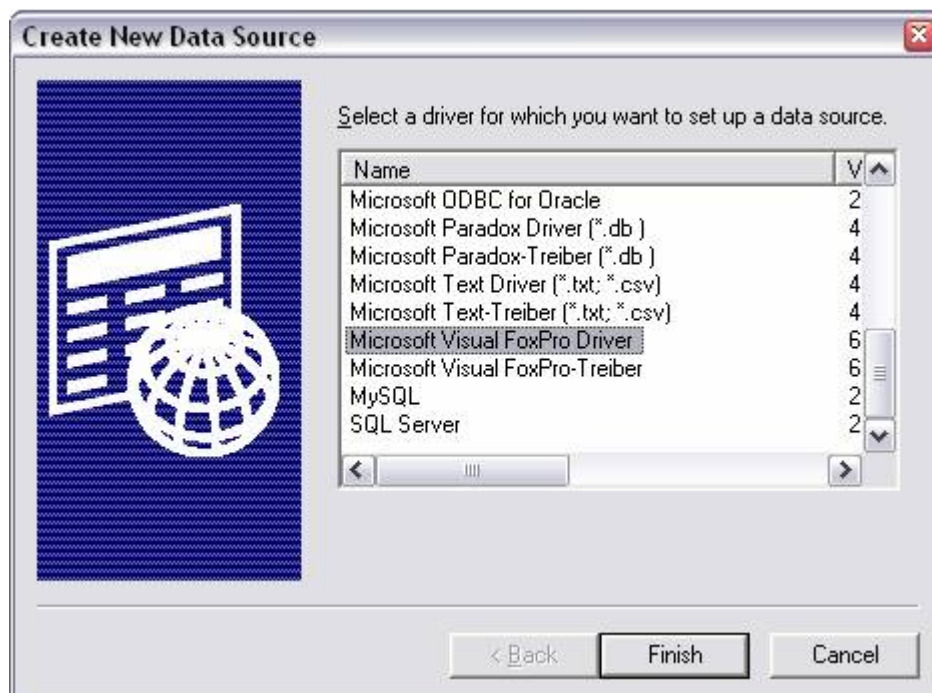


Fig. 7. Create new data source

On the interface above, the highlighted driver as it is in the diagram is to be selected and then click finish. Another window appears for the Visual FoxPro set up. Remember that all that was done above is just to create a new data source and now it should be set up or configured. The window given below in figure 8 is the VFP ODBC set up.

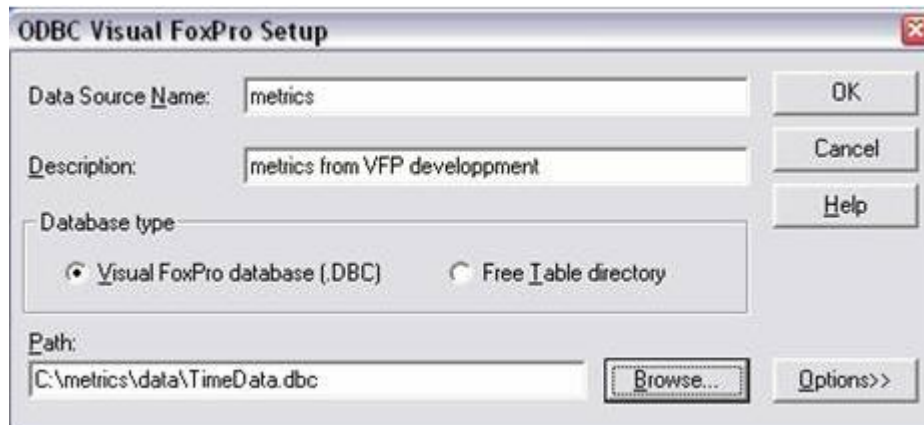


Fig. 8. Visual FoxPro ODBC setup

A closer look at the interface given by figure 8 provides two options to choose from in the database type section. The first option which is the “Visual FoxPro database (.dbc)” is selected since we are working on transferring the whole database and not just the tables. A click on the OK button confirms the choice and ODBC data source administrator window appears only this time the newly created connection has been added to the list.

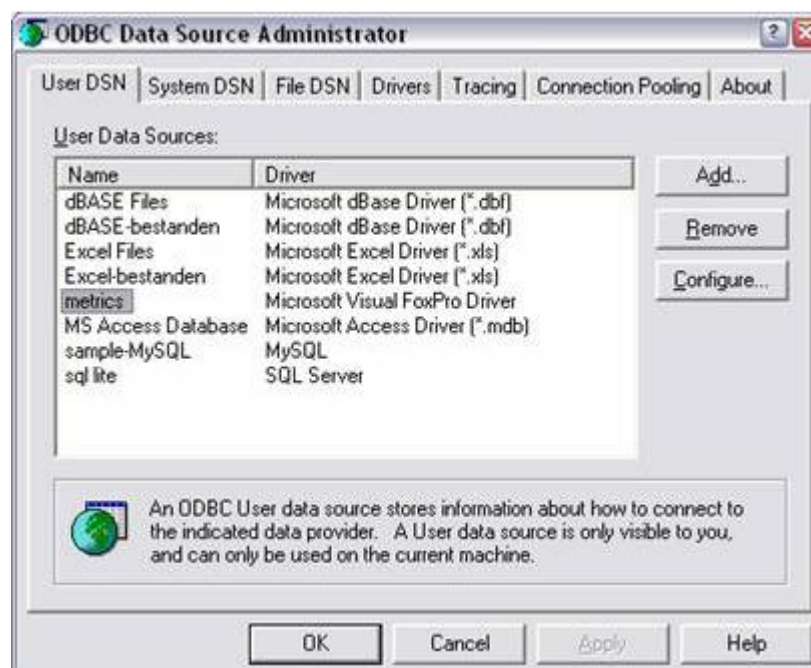


Fig. 9. ODBC source administrator

If this procedure is followed appropriately, a connection is certainly created to the intended VFP database. It should be noted that the name “meric” used in this explanation is just to present a procedure. The user can give any name as desired to when implementing a similar connection.

Since the connection has been established, the next thing is to click on the Microsoft database management icon to get it running. It uses a menu option to connect to Visual FoxPro database if configured appropriately. There is a need to create a new empty database folder where the imported VFP database will be stored. Select the “blank Access database” option given on the Access database system interface and then click OK.

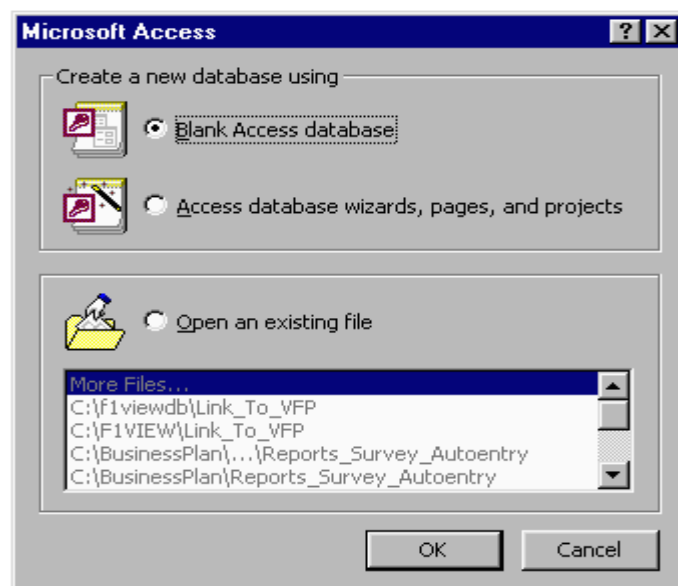


Fig. 10. Creating Microsoft Access blank database

What this simply means is that an empty folder to house the incoming data is to be created. By choosing the ok button, a second interface appears and from the file menu there is a possibility to specify the source database directory.

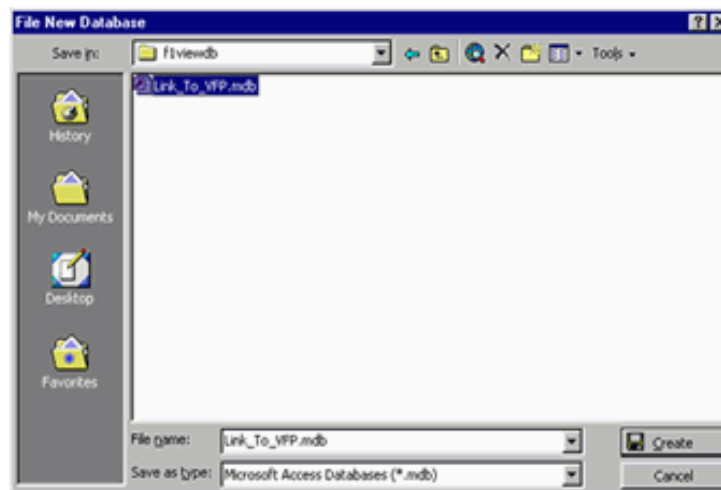


Fig. 11. Blank database

The database name in the example is “F1viewdb” and it is located in the same directory as C:\F1viewdb. An important precaution should be that the Visual FoxPro database is kept running all through the conversion process. Automatically, the file name extension becomes F1viewdb.mdb. This file extension (.mdb) is how databases are saved in Microsoft Access. By clicking OK, the new database folder is created and waiting for its data. /30/ /31/

On the Access menu, click on tab “file”, then “GET External Data” option. This action will prompt a new smaller window to appear and this is where it gets confusing.

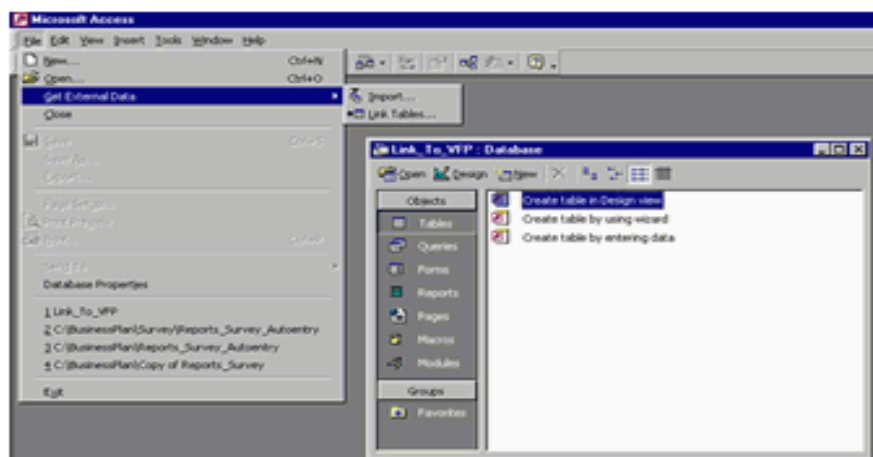


Fig. 12. Microsoft Access importing data

There is another option to choose “import” but the right option is the “Link tables”. By clicking this option, it means an updated version of the tables is read each time the VFP database is downloaded but if the Import option was chosen, and then no upgraded version of the table can be imported when changes are made. /31/

Another interface appears as soon as the “link tables” option is clicked.



Fig. 13. Linking tables

Beneath this interface is the “file name” and “files of type” options. Scroll to the last option of the “files of type” list to choose the “ODBC Database ()” option. A “Select Data Source” interface appears and right above it there are two clickable options; “File Data Source” and “Machine Data Source”.

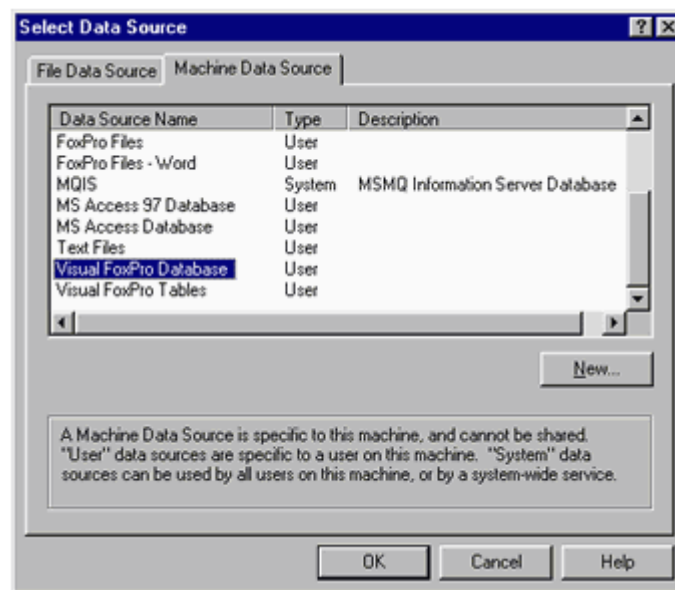


Fig. 14. Select data source

In the second option, choose “Visual FoxPro Database” from the lists available. Another tricky option is choosing the “Visual FoxPro tables” but this option is not the appropriate choice if transferring the database and not just tables. /30/

To link tables to transfer, the appropriate tables are selected from the list of tables given as shown below.

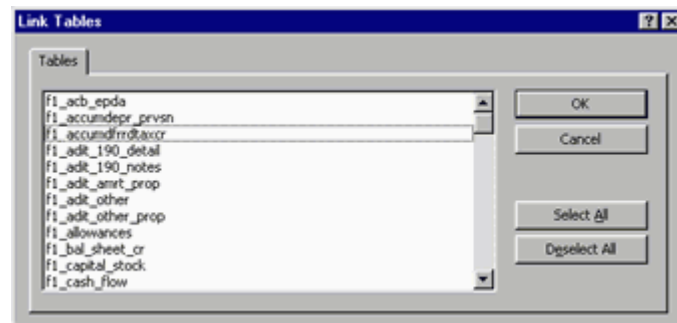


Fig. 15. Selecting tables to link

When this is done, click OK and the table(s) selected will appear on the Microsoft Access database tab with a globe icon to differentiate the tables created in Microsoft Access from the imported ones.

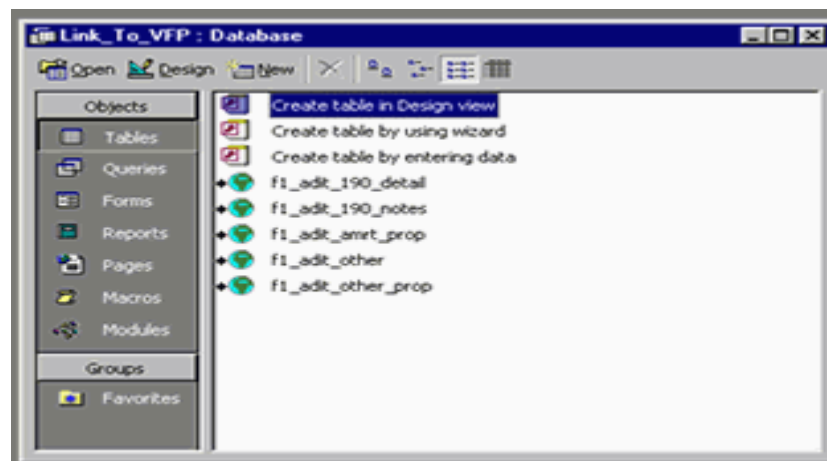


Fig. 16. Green globe to identify exported data

At this point, the imported tables are saved which means they have been converted to .mdb format and can then proceed to convert from Microsoft access to the final destination which is the PostgreSQL.

6.3. Conversion from Microsoft Access to PostgreSQL using ODBC

To begin this last phase of conversion, from Microsoft access to PostgreSQL using ODBC connection, the steps below are followed and it is worth to note at this stage that it is assumed that the two database systems have been installed and are working perfectly.

- Install PostgreSQL ODBC driver
- Create ODBC DSN

- Export from Microsoft Access database system
- Edit Access properties to link by using the PostgreSQL DSN created

6.3.1. Install PostgreSQL ODBC driver

As an open source database management system, a copy of the newest PostgreSQL ODBC driver can be easily found on the internet, downloaded and installed on the computer system. It is very important that the user verifies the computer's architecture to know if it is a 32 or 64 bit before downloading and installing the PostgreSQL driver. There are 64 bit supported versions of the ODBC and also are the 32 bit versions. /33/

6.3.2. Create ODBC DSN for PostgreSQL

DSN stands for data source name and it is one good way to access databases. In this conversion process, a link to the PostgreSQL tables and views via the link tables will be created. To do this, a similar approach to what was done in chapter 6.2.2 when creating the data source will also be followed. PostgreSQL and VFP ODBC drivers should not be misinterpreted as same but the installation process is similar and it is the same user interface. After clicking the link button a window is shown as below



Fig. 17. Creating DSN for PostGreSQL

On this link interface, click on the file of types drop down menu and scroll to the last option "ODBC Databases ()".

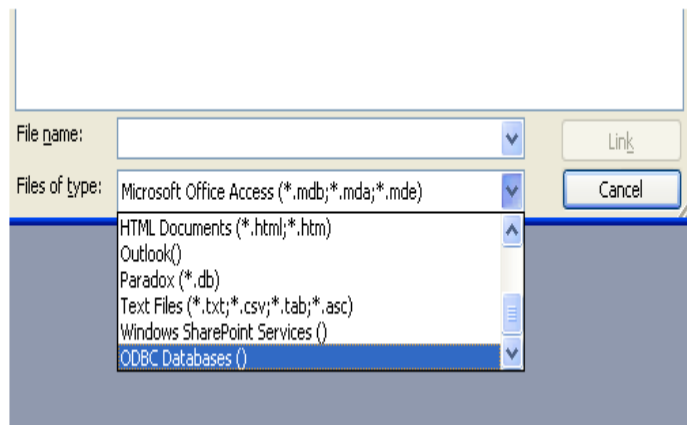


Fig. 18. Selecting the right properties for DSN

The interface continues to provide an appropriate window as soon as the OK button is clicked to confirm that the user has selected an option. During this set up, it is wise to switch to the “file data source” option instead of the “Machine Data source” and the reason is because file DSN string gets automatically embedded in Microsoft Access database so there will not be a need to set up DSN on every computer that will connect to the Ms Access database. The next thing to do is to create a new PostGreSQL DSN.

From the interface, select the “Add” button and that action automatically opens a new window known as create new data source.

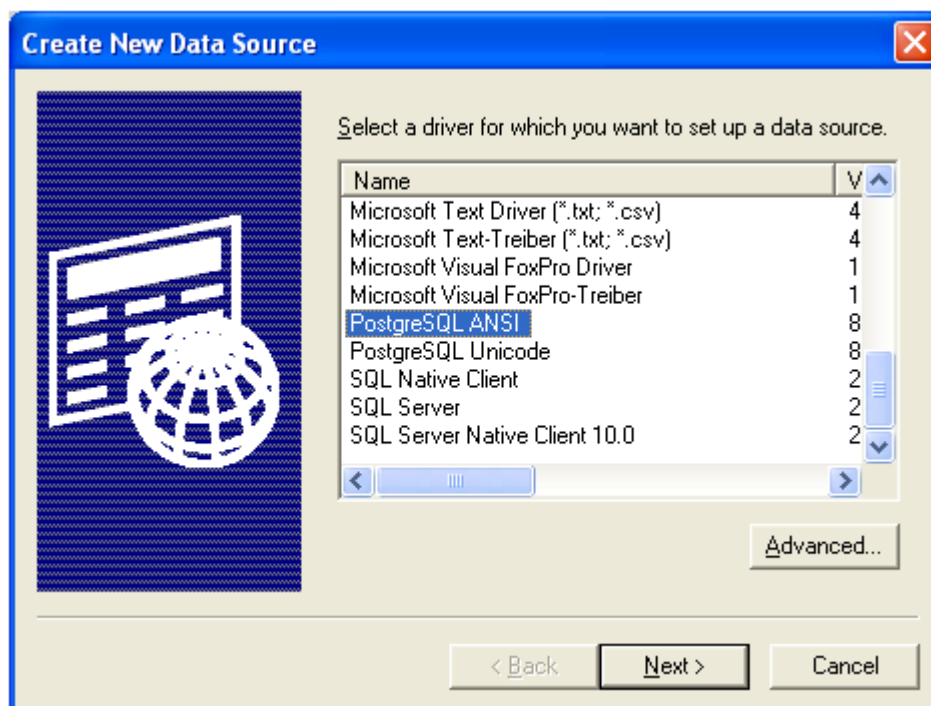


Fig. 19. Selecting driver to set up data source

PostgreSQL ANSI is the selected choice from the list as shown above in figure 19. That driver is used to create the new DSN and the ANSI part just simply shows that the database is WIN-1252 encoded. As soon as the next button is clicked, several pages appear with series of property amending options for the connection which must be filled appropriately. /34/

A click on “data source” button which can be found at the bottom on the left makes the next window appear where the setting options are made by switching between two pages. The Reader should note that the “Bools as Char” option must be unchecked and select the “Text as LongVarChar” before clicking OK. /34/



Fig. 20. Amending properties for connection

6.3.3. Export to PostgreSQL from Microsoft Access

The last part of this conversion is to transfer the database to PostgreSQL and this is done by clicking on the Microsoft Access database system. From the file menu, go to the “Export” option following a similar procedure as used in the conversion of VFP to Ms Access and then select “ODBC Data source” from the file of type. Finally, click on the PostgreSQL DSN created, select the tables, views and all that is needed to be exported and click OK. When the exporting process is complete, depending on the size of the transported database, some parameters such as the fields and table name might need to be edited. In the best practice, it is better to edit those mentioned parameters while in the Microsoft Access database to lowercase which saves the problem of having to do it after the final conversion as PostgreSQL is case sensitive. /34/

7. CONCLUSIONS

There are series of problems encountered during this research work and the first is the inaccessibility to some of the database applications. There are two commercial database systems used and by commercial it simply means that they cannot be accessed freely like the open source applications unless paid for. The only option is to search the internet for materials to use. Unfortunately, there are not so many detailed books written especially on one of the database application.

The major challenge, the second, is the complexity of the project. The problem stipulates conversion between two database systems but the solution makes use of an additional database application making it three database applications to analyze. This made the research more tensed and tasking than initially assumed.

The third challenge, a triangular problem is to strike a balance between gathered resources, residual knowledge of the database system and writing an easily comprehensible document was a bit hectic at the beginning. As it appeared, developers in the database industries paid less attention to the source database application used in this project and preferred to write more on open source product since it is easy to get and install for use.

Aside those problems stated above, this research work is written with the intension to enlighten the reader of this document by analyzing the data migration concept and providing clarity on misconceptions attached. Perceiving from the reader's view, all the database systems used was introduced; explained migration before commencing to the real deal of converting from one database to another. There are obviously other ways to go by this conversion such as manual coding but to do that, certain tools must be provided and all database systems must be made available, which is very costly considering all the tools and testing gadgets. However, the conversion approach proposed and used in this document recorded a success in implementation, therefore serves as a foundational document on which a bigger data migration project can be built.

8. REFERENCES

- /1/ NetApp Global Services, Data migration best practices, [PDF-document], <http://partners.netapp.com/go/techontap/NGS_migration.pdf> 17.12.2011.
- /2/ Data migration [WWW-document], <http://www.infotechnet.org/ntca/DataMigration.htm#_DoWeMigrate> 17.12.2011.
- /3/ John Morris, Practical data migration, [WWW-document], <<http://www.amazon.ca/Practical-Data-Migration-John-Morris/dp/1902505719>> 18.12.2011.
- /4/ Introduction to Visual FoxPro, [PDF-document], <<http://www.tedroche.com/Present/2001/PRE201.pdf>> 21.12.2011.
- /5/ What is Visual FoxPro, [WWW-document], <<http://www.craigbailey.net/what-is-vfp/>> 22.12.2011.
- /6/ Visual FoxPro programming standards and guidelines, [PDF-document] <<http://www.whitelightcomputing.com/resources/WLCDeveloperGuidelines4point0.pdf>> 21.12.2011.
- /7/ Databases in Visual FoxPro, [WWW-document], <[http://msdn.microsoft.com/en-us/library/w87s2tx4\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/w87s2tx4(v=vs.80).aspx)> 22.12.2011.
- /8/ Visual FoxPro File Extensions and File Types, [WWW-document], <<http://support.microsoft.com/kb/130686>> 22.01.2012.
- /9/ Creating tables, [WWW-document], <[http://msdn.microsoft.com/en-us/library/zchks17\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/zchks17(v=vs.80).aspx)> 22.01.2012.
- /10/ Visual FoxPro data and field types, [WWW-document], <[http://msdn.microsoft.com/en-us/library/aa975357\(v=VS.71\).aspx](http://msdn.microsoft.com/en-us/library/aa975357(v=VS.71).aspx)> 26.01.2012.
- /11/ PostgreSQL 9.0.7 Documentation, [PDF-document], <<http://www.postgresql.org/files/documentation/pdf/9.0/postgresql-9.0-A4.pdf>> 29.01.2012.
- /12/ PostgreSQL, [WWW-document], <<http://www.postgresql.org/about/>> 2.01.2012.
- /13/ PostgreSQL: Introduction and concepts, [PDF-document], <http://zid-luxinst.uibk.ac.at/postgresql/aw_pgsql_book.pdf> 29.01.2012.
- /14/ Packaging Related Objects into an Extension, PostgreSQL, [WWW-document], <<http://www.postgresql.org/docs/9.1/static/extend-extensions.html>> 02.02.2012.

/15/ Database File Layout, POSTGRESQL Documentation, [WWW- document], <<http://www.postgresql.org/docs/8.1/static/storage.html>> 05.02.2012.

/16/ PostgreSQL Home Page, Data types, <<http://www.postgresql.org/docs/7.4/static/datatype.html>> 07.02.2012.

/17/ Simple Access, What is Microsoft Access, [WWW-document], <<http://www.simply-access.com/What-Is-Microsoft-Access.html>> 08.02.2012.

/18/ Microsoft Access Versions,[WWW-document], <<http://www.kellermansoftware.com/t-articleaccessversions.aspx>> 08.02.2012.

/19/ Introduction to Microsoft Access, [PDF-document], <<http://www.ischool.utexas.edu/technology/tutorials/office/access03/access2003.pdf>> 10.02.2012.

/20/ Introduction to Microsoft Access, [WWW-document], <<http://www.functionx.com/access/Lesson01.htm>>09.02.2012.

/21/ OpenGate software, Microsoft Access Version Comparison, [WWW-document], <<http://www.opengatesw.net/ms-access-tutorials/Microsoft-Access-Version-Comparison.htm>> 10.02.2012.

/22/ Microsoft Access 2010 in Depth, [WWW-document], <<http://my.safaribooksonline.com/book/databases/microsoft-access/9780768695229>> 10.02.2012.

/23/ Access programmers, Microsoft File Extensions, [WWW-document], <<http://www.access-programmers.com/file-extensions.aspx>> 14.02.2012.

/24/ Microsoft Access Table Design, [WWW-document], <http://www.databasedev.co.uk/fields_datatypes.html> 17.02.2012.

/25/Microsoft office Home page, Introduction to data type and field properties, [WWW-document], <<http://office.microsoft.com/en-us/access-help/introduction-to-data-types-and-field-properties-HA010233292.aspx>> 17.02.2012.

/26/ Microsoft Home page, How to Import and Export Access tables with DAO or ODBC, [WWW-document], < <http://support.microsoft.com/kb/200427>> 20.02.2012.

/27/ IBM Global Technology Services, Best practices for data migration, [PDF-document], <<http://www-935.ibm.com/services/us/gts/pdf/softek-best-practices-data-migration.pdf>> 20.02.2012.

/28/ Cary Jensen and Loy Anderson, Advantage Database Server, Advantage and Visual FoxPro, [PDF-document], <<http://www.jensendatasystems.com/ADSBook10/files/Chapter%2020%20Advantage%20and%20Visual%20FoxPro.pdf>> 21.02.2012.

/29/ Microsoft Support Home page, Link to a FoxPro Database, [WWW-document], <<http://support.microsoft.com/kb/824264>> 26.02.2012.

/30/ Federal Regulatory Energy Commission, Connecting to a Visual FoxPro Database through Microsoft Access, [WWW-document], <<http://www.ferc.gov/docs-filing/forms/ms-access.asp>> 26.02.2012.

/31/ Boudewijn Lutgerink, Accessing your VFP Database in Access, [WWW-document], <<http://www.foxite.com/articles/read.aspx?id=32&document=accessing-your-vfp-database-in-access>> 28.02.2012.

/32/ Microsoft Support Home page, ODBC connectivity overview, [WWW-document], <<http://support.microsoft.com/kb/110093>> 28.02.2012.

/33/ Microsoft support Home page, How to create a DSN in windows 7, [WWW-document], <<http://support.microsoft.com/kb/305599>> 02.03.2012.

/34/ Regina Obe and Leo Hsu, Using MS Access with PostGreSQL, [WWW-document], <<http://www.postgresonline.com/journal/archives/24-Using-MS-Access-with-PostgreSQL.html>> 04.03.2012.

