Design and Implementation of Charcon Algorithm to Solve Arabic Database Character Set Migration Corruption Focus On: Migrating Arabic Data to Different Character Set

[1]Majzoob K. Omer, [2]Adel A. Ahmed and [2]Mojtaba Eltayeb
[1]*Al-Neelain University, Khartoum, Sudan*
[2]*Red Sea University, Khartoum, Sudan*

Key words: Oracle database, Character set, Charcon, Arabic character, recovery

Abstract: This study aims to allow upgrading oracle database with arabic data to different character set without data loss is required. The character set converter mechanism Charcon is designed to solve this problem, Charcon which is a module will allow oracle database to deal with arabic character set migration operation, it allows moving arabic data between different character set quickly, safely and efficiently, Charcon uses a mechanism that protects the Arabic data in Flat file internally and executes the conversion process then restore the data to its original location, it has many modules, each one each one has its own function, it supports checking conversion possibility, executing the conversion process and restoring the data in addition to data protection it allows cancelling the conversion operation at any time without any data loss. The manual stimulation of Charcon procedures works perfect in real database, without data loss or corruption only some consideration noticed to optimize the conversion process, starting with performance issue while converting big database with millions of rows it may take a long time depending in data size also it may consume machine resource if multiple instances of Charcon method is running at the same time in addition to the recovery consideration while the process is running and interrupted, the method should allow data recovery without loss.

Corresponding Author:
Majzoob K. Omer
Al-Neelain University, Khartoum, Sudan

INTRODUCTION

Upgrading IT environment for any business that uses Information Technology is an important process that may occur every couple of period, the upgrade process may require changes in existing applications and database, the database itself as a backbone of IT infrastructure will require upgrade either to expand and increase its capabilities and improve its performance or for solving problems[1].

	0	1	2	3	4	5	6	7	A	B	C	D	E	F
0	NUL	DLE	SP	0	@	P	`	p	NBSP	°	À	Ð	à	ð
1	SOH	DC1	!	1	A	Q	a	q	¡	±	Á	Ñ	á	ñ
2	STX	DC2	"	2	B	R	b	r	¢	²	Â	Ò	â	ò
3	ETX	DC3	#	3	C	S	c	s	£	³	Ã	Ó	ã	ó
4	EOT	DC4	$	4	D	T	d	t	¤	´	Ä	Ô	ä	ô
5	ENQ	NAK	%	5	E	U	e	u	¥	µ	Å	Õ	å	õ
6	ACK	SYN	&	6	F	V	f	v	¦	¶	Æ	Ö	æ	ö
7	BEL	ETB	'	7	G	W	g	w	§	·	Ç	×	ç	÷
8	BS	CAN	(	8	H	X	h	x	¨	¸	È	Ø	è	ø
9	HT	EM	)	9	I	Y	i	y	©	¹	É	Ù	é	ù
A	NL	SUB	*	:	J	Z	j	z	ª	º	Ê	Ú	ê	ú
B	VT	ESC	+	;	K	[	k	{	«	»	Ë	Û	ë	û
C	NP	FS	,	<	L	\	l	|	¬	¼	Ì	Ü	ì	ü
D	CR	GS	-	=	M	]	m	}		½	Í	Ý	í	ý
E	SO	RS	.	>	N	^	n	~	®	¾	Î	Þ	î	þ
F	SI	US	/	?	O	_	o	DEL	¯	¿	Ï	ß	ï	ÿ

Fig. 1: ISO 8859-1 8-Bit encoding scheme

Oracle Database as the most widely used databases in the world, may require changes in its infrastructure to meet the upgrade process, either because of application upgrade to new version or to support new features and languages[2].

Oracle database stores data using character set for its default template, the upgrade process may require changes to Oracle database default character set itself, here the database administrator will face the problem while they are trying to make change to the database default character set, the database didn't support changing the current default character set which is WE8MSWIN1252 to the new character set which is AR8MSWIN1256 that is required by the new upgrade and if the database administrator insist to execute the Conversion then all the existing Arabic data will damage and the operation is not recoverable, the available conversion mechanism available for Oracle database is not able to change the character set between two character set that both of them uses 1 byte to store letter weather it is English letter or Arabic, the available mechanism only allow migration to Superset Character set that is different than the Character set required by the running applications[3].

An Oracle database is a product of Oracle Corporation which is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is a key to solving the problems of information management[4].

Oracle database is the first database designed for enterprise grid computing, the most flexible and cost-effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads because capacity can be easily added or reallocated from the resource pools as needed[5].

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

Oracle Database enables to store, process and retrieve data in native languages. It ensures that database utilities, error messages, sort order and date, time, monetary, numeric and calendar conventions automatically adapt to any native language and locale[6]. (Oracle Database Globalization Support Guide page 1-1).

When computer systems process characters, they use numeric codes instead of the graphical representation of the character as shown in Fig. 1. For example, when the database stores the letter A, it actually stores a numeric code that is interpreted by software as the letter. These numeric codes are especially, important in a global environment because of the potential need to convert data between different character set. Needs to convert data between different character set to support additional languages or to solve application compatibility issues is very important, especially with large databases, oracle database didn't support direct migration to non-superset character set which will cause data corruption but if the database character set uses a single-byte character encoding scheme, no data loss occurs when characters are stored because the number of characters is equivalent to the number of bytes. If the database character set uses a Multibyte character set, then the number of bytes no longer equals the number of characters because a character can consist of one or more bytes. (oracle database Globalization Support Guide 10g Release 1, page 2-1).

Character set: When computer systems process characters, they use numeric codes instead of the graphical representation of the character. For example, when the database stores the letter A, it actually stores a numeric code that is interpreted by software as the letter.

Table 1: Single-byte and multibyte encoding

Characters	WE8MSWIN 1252 encoding	AL32UTF8 encoding
ä	E4	C3 A4
ö	F6	C3 B6
©	A9	C2 A9

These numeric codes are especially important in a global environment because of the potential need to convert data between different character sets[7].

Multi-byte storage requirement: During migration to a new character set, it is important to verify the column widths of existing CHAR and VARCHAR2 columns because they may need to be extended to support the encoding that requires Multi-byte storage. Truncation of data can occur if conversion causes expansion of data[8]. Table 1 shows an example of data expansion when single-byte characters become multi-byte characters through conversion.

MATERIALS AND METHODS

Environment setting: Another reason leads to data corruption is the wrong set of NLS_LANG parameter, when the parameter is not set properly on the Client. The NLS_LANG value should reflect the client operating system code page. For example, in an English Windows environment, the code page is WE8MSWIN1252. When the NLS_LANG parameter is set properly, the database can automatically convert incoming data from the client operating system. When the NLS_LANG parameter is not set properly, then the data coming into the database is not converted properly. For example, suppose that the database character set is AL32UTF8, the client is an English Windows operating system and the NLS_LANG setting on the client is AL32UTF8. Data coming into the database is encoded in WE8MSWIN1252 and is not converted to AL32UTF8 data because the NLS_LANG setting on the client matches the database character set. Thus oracle assumes that no conversion is necessary and invalid data is entered into the database[9].

This can lead to two possible data inconsistency problems. One problem occurs when a database contains data from a character set that is different from the database character set but the same code points exist in both character sets. For example, if the database character set is WE8ISO8859P1 and the NLS_LANG setting of the Chinese Windows NT client is SIMPLIFIED CHINESE_CHINA.WE8ISO8859P1, then all Multibyte Chinese data (from the ZHS16GBK character set) is stored as multiples of single-byte WE8ISO8859P1 data. This means that Oracle treats these characters as single-byte WE8ISO8859P1 characters, the second problem is that can cause the loss of data is migrating a database that contains invalid data. Invalid data usually occurs in a database because the NLS_LANG parameter

is not set properly on the client. The NLS_LANG value should reflect the client operating system code page. For example, in an English Windows environment, the code page is WE8MSWIN1252. When the NLS_LANG parameter is set properly, the database can automatically convert incoming data from the client operating system[13]. When the NLS_LANG parameter is not set properly, then the data coming into the database is not converted properly. For example, suppose that the database character set is AL32UTF8, the client is an English Windows operating system and the NLS_LANG setting on the client is AL32UTF8. Data coming into the database is encoded in WE8MSWIN1252 and is not converted to AL32UTF8 data because the NLS_LANG setting on the client matches the database character set. Thus Oracle assumes that no conversion is necessary and invalid data is entered into the database.

So, migrating database to different character set is a difficult operation and restricted to only limited choices, these choices have performance implication in addition to the incompatibility issue with some user end application, for those two reasons migrating to another character set needs a well-prepared procedures and using many external tools to do this Operation, this happen in normal circumstance, the situation will be more complicated when we have Arabic data in the database because Arabic has different writing direction and different script and it is only supported with small number of character set groups and most of these character set has compatibility issue with application development.

A possible scenario assumes that an organization in Sudan has a large size database with character set WE8MSWIN1252 setting and this organization uses Oracle Forms 6i as end-user application the Organization is planning to move to web tier which requires migrating the end user application to forms 11 g, the forms 11 g require different character set which is AR8MSWIN1256 to work well and this situation needs a compatible database character set which is AR8MSWIN1256, unfortunately in this case. The migration will be nightmare because oracle database didn't allow migrating between those two types of character set and if the database administrator insists to run the migration commands against the database then all Arabic data will be damaged and non-recoverable, so, finding a solution that allows oracle database to support this type of migration will be a great support for organization using Arabic language to move forward to any character set that matches their needs whatever was the reason of migration (Fig. 2).

Character set conversion requirement: In some circumstances the main database requires changes to allow the new upgrade to the database applications, these changes may include changing the default database
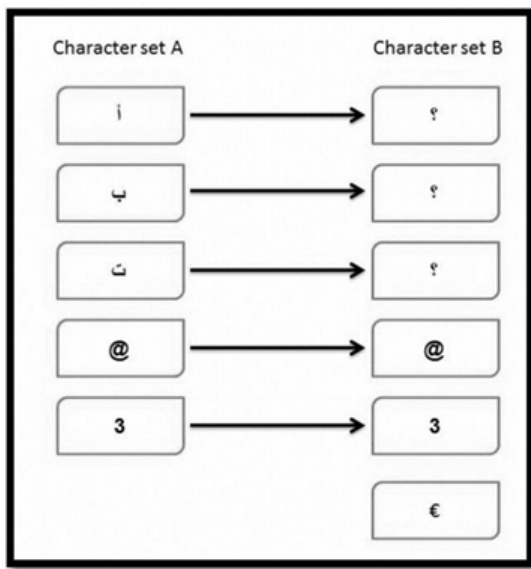
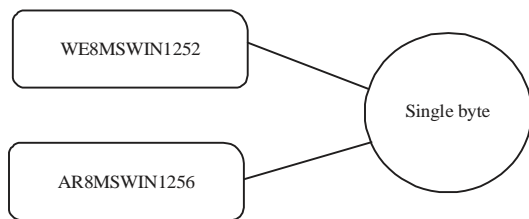Fig. 2: Character set conversion corruption sample

Fig. 3: Shows both character set is single-byte character set

character set for all the existing, the main two requirements that require database character set changes are:

To Support new characters: Different character set contains different groups of characters as mentioned earlier, for example, Fig. 3 represents the ISO-8859p1 Character set but this collection didn't contains Arabic characters, so, if an organization uses applications that use this character set and need to store Arabic characters they need to migrate their character set to different one than contain the Arabic characters in addition to the rest characters or if using character set that didn't contain the Euro symbol and the application requires it, here the system must migrate to a new character set that supports the new character, so, it is very important to have the ability in the feature to migrate to different character set because of the need to support new characters.

To upgrade application to a new version: Database application or user end application has compatibility

issues with database character set, this means in special circumstance if application is planning to use new release of database application you must make sure that the new release is compatible with your existing character set for example if your existing systems is using Oracle Forms 6i and you need to migrate to 3 tier model or web based Oracle forms 11 g which require Unicode

UTF8 character set and your current configuration is using different character set, so in this case you need to migrate your character set to enable migration to new application release[10].

So, there is a huge need for character set migration with small size database this may not cause problems because this can be solved by re inserting the data from scratch after changing the character set setting to a new one but if re inserting the data is not possible because of the amount of the data then the organization will face a serious problem because oracle database didn't allow Direct character set migration to a non-superset character set or even to non-logically compatible character set and if database enforced to execute the character set Conversion then all existing Arabic data will be corrupted and that corruption is not recoverable, so, this thesis is intended will give the oracle database the ability to automatically allow and execute this migration without data corruption or truncation.

Character set converter module: AS Mentioned earlier In Oracle, a database character set defines a set of characters that can be stored in the database and defines how each character maps to a particular sequence of bytes stored in memory or disk to represent this character. What Oracle calls a character set is also known in the industry as a character encoding which can be thought of as a set of characters plus their encoding in bytes, database character set may be single-byte where each character has one byte or multibyte varying width where each character may be 1-4 bytes with the maximum dependent on the particular character set. Single-byte character sets are easier and faster to process but have a very limited capability of 256 codes too few even for all European languages. Multibyte character sets require more complex string processing but can accommodate thousands of characters.

Almost all character sets other than Unicode (frequently called legacy character sets) were each designed for a specific language or group of languages. For single-byte character sets, this was logical because of the limited number of available codes. For multi byte character sets, this was presumably because of limited requirements of isolated local databases, the desire to limit the maximum required a width of a character code or expertise required to design a character set for multiple unrelated language groups.

Now the idea is to find a safe way to migrate database character set WE8MSWIN1252 which is

Fig. 4: Arabic data inserted and retrived

European character set to AR8MSWIN1256 which is Arabic Latin character set, this conversion is not possible and lead to data corruption as Oracle Said because the target character set which is AR8MSWIN56 is not superset from the source but in the Practical demonstration the representation of Arabic character is done using the character set WE8MSWIN1252 as the target character set AR8MSWIN1256 but the database is unable to move the character correctly, it is a Database Mechanism Fault (DMF) because the corruption happens because of the mechanism itself not for other reason.

Principle of Charcon: Both are legacy character set that means both character set related to single-byte character set as shown in Fig. 3 that mean the length of any character in both groups is single byte = 8 bits or one character, so no double byte here and all characters in each group have code between 0 and 127, so every group can represent the other group as long as the machine will not damage the data, here it appears that a suitable way to move the data from the first character set (WE8MSWIN1252) to the target character set (AR8MSWIN1256) is needed and this method will be executed by the database itself automatically: Now to simplify the operation let's assume that:

- X: is the Arabic string that needed to move between the two character sets
- A[]: is the location to store the string with the WE8MSWIN1252 template
- B[]: is the location to store the string with the AR8MSWIN1256 template
- An [X] means there are Arabic data stores in the Location A with WE8MSWIN1252
- B[X] means there are Arabic data stores in the location B with AR8MSWIN1256
- So my operation should achieve the following
- A[X] = B[X], A[X]–B[X] = 0

To achieve this, a new store location called C[ ] is declared which is Unicode text flat file. This file will be used by the Module to bypass the Database Mechanism Fault (DMF) and will help us to achieve the goal. So, the operation will be:

- Set A[X]
- Move A[X] to C[ ], now C[ ] become C[X]
- Move C[X] to B[X]

In the first step, Arabic data is inserted which is X in the A location with WE8MSWIN1252 character set, this is normal database insert SQL command. Insert into Master (id, name, admin) values (1,' محمد ا حمد ', my'); Insert into Master (id, name, admin) values (2,' , الاحمر البحر'Y'); commit. Now, there is Arabic data in location (A) with Western single byte Unicode (WE8MSWIN1252) at this step, if we query the data using the following SQL statement: Select * from Master; the rows will show Arabic data retrieved from the table master as shown in Fig. 4.

In the second step, Arabic data needs to move to C[] which is Unicode text flat file to do this step we need to translate it to various step in the real life. The next step is simulating the second step which includes:

- Read the Arabic data location
- Extract the data to the Unicode FLAT files using metadata
- Empty or delete the location of the data from the existing data
- Prepare the new location

The above procedure will be executed manually to ensure that it will return the same result will be returned by the database process created for this step. The third step which is Moving C[X] to B[X] will move back the data from the Unicode flat file to its original location with new Character set setting need to be used which is AR8MSWIN1256.
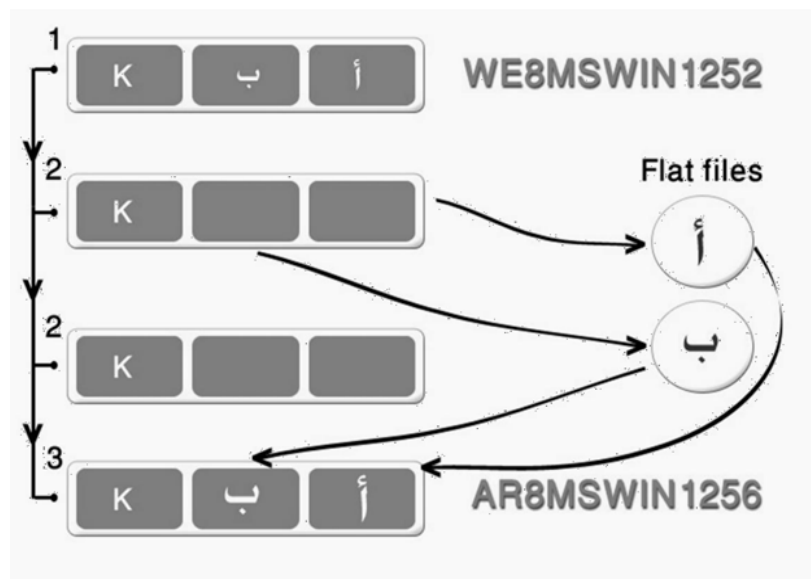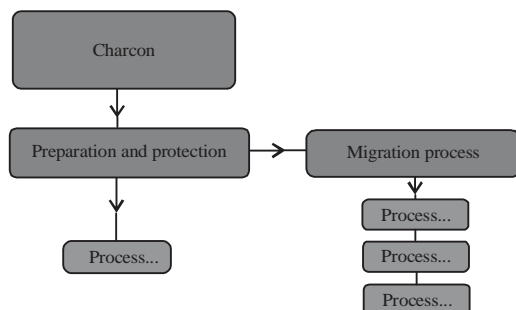
Fig. 5: The idea of flat files

Fig. 6: Show the main two function of the Charcon module

Figure 5 show very basic concept of the solution to solve the corruption while the extraction of the concept above will lead to more complicated design when converted to Software concept in the coming steps.

How character set converter works: To implement each step must be implemented Oracle database, without neglecting any small detail, taking consideration of all database architecture and instance details including background processes, memory structure and software lifecycle, the solution design will divide into two functions:

- Prevent characters set corruption
- Execute the Conversion process safely

As shown in Fig. 6 after implementing the solution the Module will be responsible for two procedure, preparing for database migration and protecting the data from any possible corruption that may occur whether by forwarding the execution process to another module responsible for migration process or migrate the data directly safely after ensuring that the corruption will not occur.

The Charcon (character converter) software is a module will be implemented as a part of the oracle database software function, it will be responsible for all database migration and data corruption, it will be started automatically as soon as the database administration issue the migration or conversion command.

Nowadays when Oracle database administrator needs to migrate the database to any character set he issues the command:

Alter database characterset internal use <charset_name>: The above command will directly convert database character set by changing the character template using the internal mechanism, the process will not take care of the compatibility or whether the mechanism will be used to migrate the data to the new character set is suitable with the conversion or not. The implementation of the Charcon module will assume the administration will issue the coming command instead of the previous one (Fig. 7 and 8).

Alter database characterset internal use <charset_name> using Charcon: The solution can be implemented within the first basic command by making the command calling the Charcon subprogram and parse the execution to it. or it can be as a supplementary solution, it will act as migration assistant and it will be started automatically with the basic conversion command without giving the administrator the option to use it or
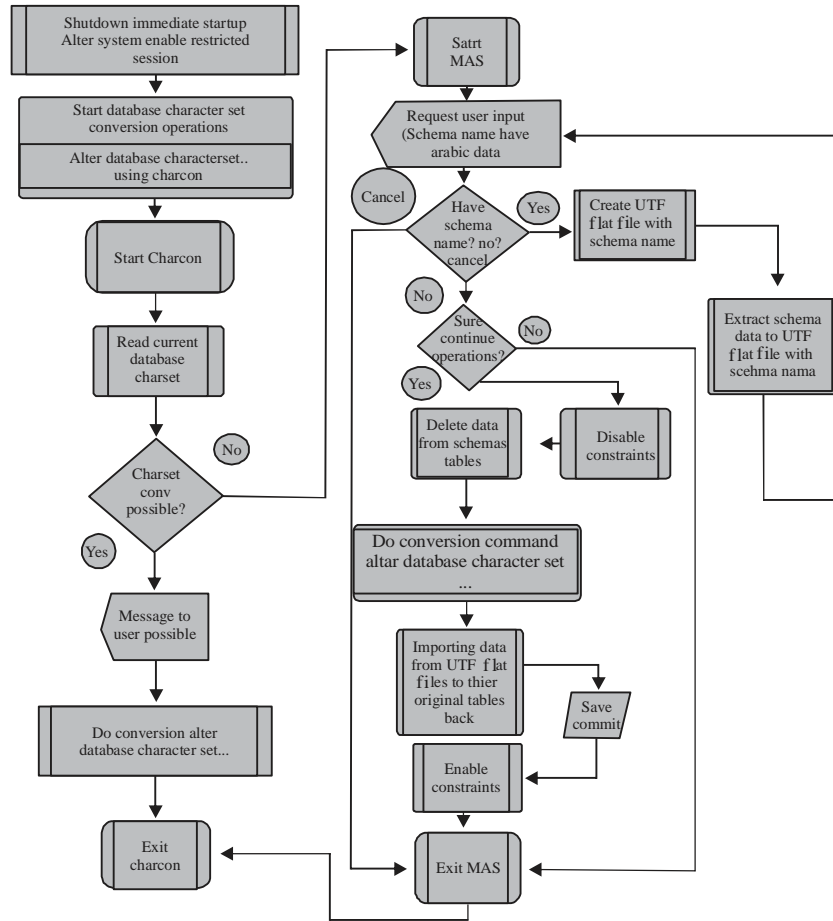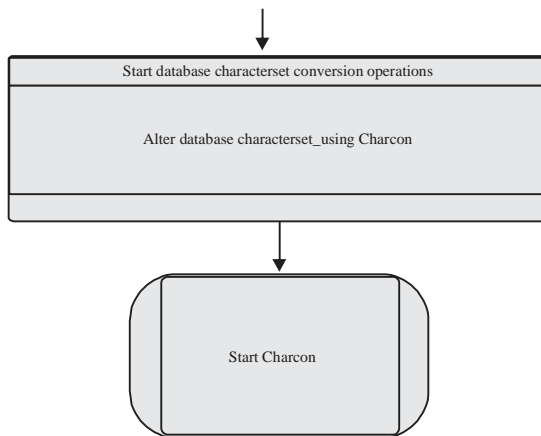
Fig. 7: Charcon big picture

Fig. 8: Charcon triggered

not, this is the perfect implementation for the solution because it will ensure that no corruption will happen.

The other option is to give the admin the option whether to use the Charcon or not and that will be by adding the keyword (using Charcon) at the end of the command as mentioned previously. Charcon formation is a multi-nested process program, the main two part is:

- The Charcon module
- The MAS module

The basic function of the Charcon module is to:
- Check current database characters set as shown in figure
- Check whether the Conversion is possible or not

If the conversion is possible (Migration to superset character set) it will alert the user that the migration is possible and if he accepts the message it will do the conversion and exit.

If the conversion is not possible it will start the MAS module which is responsible for the migration to a non-superset character set. The basic function of the MAS module is to:

- List schemas in the database with Arabic data
- Create a temporary flat Unicode file for every schema have Arabic data
- Extract schema data to the step using metadata package
- Repeat the previous step until all Arabic schemas extracted
- Disable constraints in extracted schema's tables
- Delete schema data from the database
- Issue database character set conversion command
- Read flat files and import data to their schemas back
- Enable affected schemas constraints back
- Exit MAS and call Charcon exit

The Charcon module has two functions itself to do the conversion in the first case when the target character set is superset and to start the MAS module in the second case when the target character set is non-superset as shown in Fig. 6.

Charcon data flow diagram: The basic principle of the Charcon Method is the dump and restore the Arabic data using internal mechanism provided by the database software module, it allows locating the data and dumping it and restore later using temporary placeholder with multi-character set support, the module will be responsible for triggering the conversion procedure which will change the character set for the database itself and later will execute the reverse procedure to swab back the data to its original location.

The design of the module will take the basic procedure of converting database character set as step one since the first step will require the database administrator to restrict database access and trigger the conversion procedure. The coming line will convert the database character set and it will damage the Arabic data in Database:

Alter database character set internal_use AR8MSWIN1256: From this step, the Charcon will be triggered to protect the Arabic data and to provide a safe conversion, so, the modification to the previous SQL command 4.

This will start the Charcon module internally as shown in Fig. 8 within the database in the first step it will check for current database character set and will compare it with the target character set if the character set is superset or as oracle mentioned Safe for migration then the Charcon module will allow the character set conversion as shown in Fig. 8 and if the conversion is marked impossible by Oracle due to character set incompatibility or migrating to non-superset character set, here the Charcon module will start multi-function module called MAS (Migration Assistant), the main goal for the
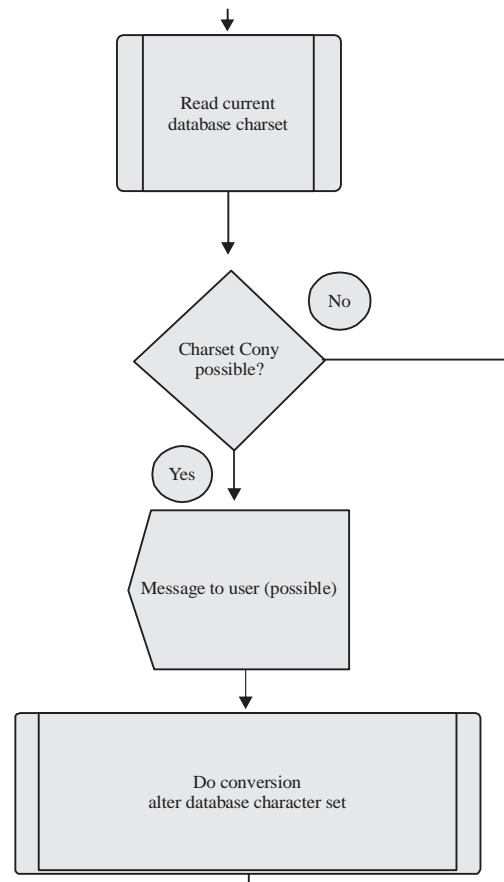

Fig. 9: Charcon data test

MAS is to protect the data and execute loop of methods to back up the data, convert the character set and restore the data as shown in Fig. 9.

MAS will be responsible for complicated operation with the help of the main Charcon, it will start executing its procedures by asking the database administrator about schemas that have Arabic data, MAS module will assume the DBA response will be one of three:

- Will fill in schema name
- Will input no schema
- Will cancel the operation

Based on the above response the MAS will take the appropriate action. in case A it will take the name of the Schema that has Arabic data and it will ask if there is another schema if no another schema is entered the MAS will create a UTF flat file with the schema name as a prefix and will extract schema DDL and data to the file in the following the order (Fig. 9).

For case B-if the DBA response with no option, the MAS will continue the conversion operation with the
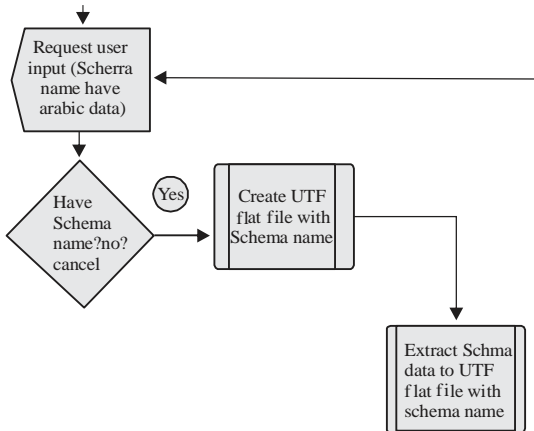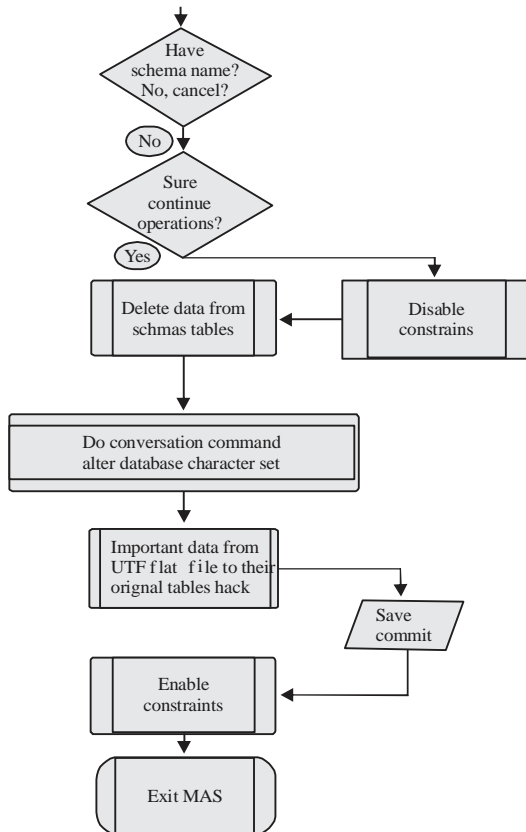
Fig. 10: MAS data extraction
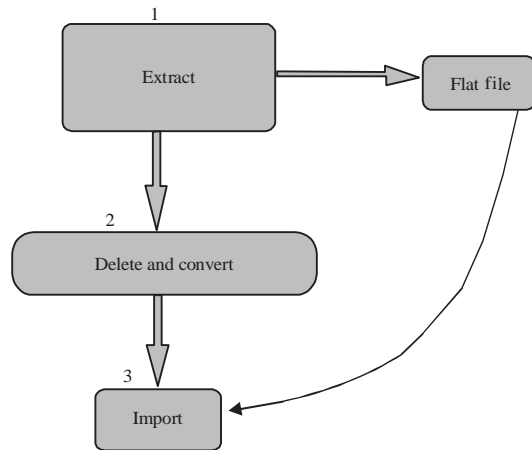

Fig. 11: MAS data conversion

Fig. 12: Dump and restore concept used by Charcon

As mentioned earlier the method and the module will protect the database and solve the corruption by executing the dump and restore principle ignoring the mechanism fault in Oracle database which causes the Arabic data corruption in spite of its ability to make the safe conversion (Fig. 11 and 12).

RESULTS AND DISCUSSION

In order to protect Arabic data conversion against corruption and to help in automatic data migration without the need for long manual procedures, the Charcon method concept should be integrated in to Oracle database operation logic and it must act as a part of the database internal operation of course, this need the modification in the Oracle database Software lifecycle itself which is not possible to implement by others, we will simulate the activity in a real database and will get the same result as if the Charcon concept implemented, the manual simulation of Charcon was implemented in a sample Corruption scenario in the real world and the result was a Perfect success of the method, we will follow the execution and will list the results.

Charcon followed dump and restore technique, it moves the data to new storage area with special setting and change the setting of the original data location then it swap back the data to its, Original location as shown in Fig. 11, Oracle database 9i with Arabic schema with the name Charcon is prepared, the schema has a sample table called Val, the database setting is WE8MSWIN1252 which is a single byte character set which can represent Arabic in Windows environment.

To simulate Charcon the procedure should manually follow the dump and restore concept as mentioned above, first a database with Arabic data using WE8MSWIN1252 character set and then it extracts the Arabic schema data to flat file using third party tool, after that it deletes the

previously chosen Schemas by disabling constraints, deleting data from specific tables, converting database character set, importing data back and enabling constraints, then it exit and return the execution to Charcon as shown in Fig. 10.

For case C-if the DBA response with cancel, this will make the MAS cancel the conversion and extraction procedures and exit and return the control to Charcon for case C the database will remain unchanged.
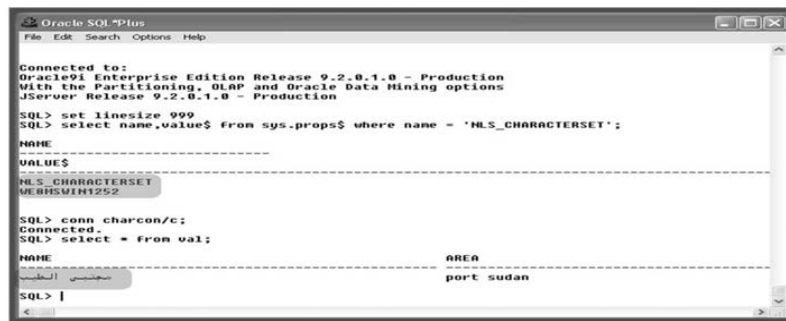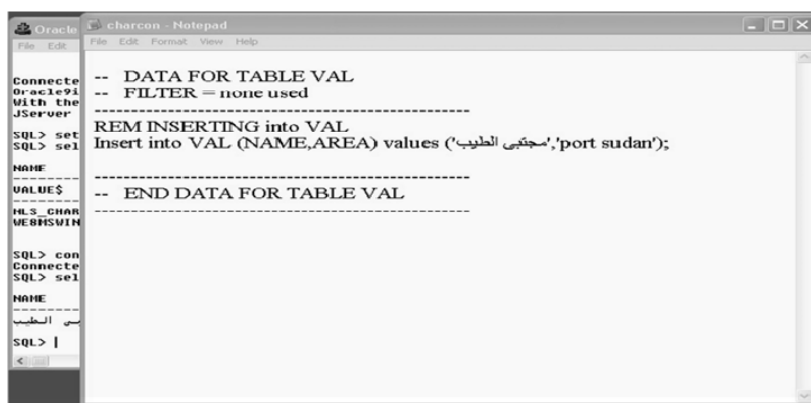
Fig. 13: Arabic data in WE8MSWIN1252

Fig. 14: Metadata creation

tables data, then it changes the database character set and at last it imports the data from the UTF flat file that prepared in the previous step.

First, a database is prepared with the data as shown in Fig. 12, the table value in the Charcon Schema hold the Arabic data needed to keep safe while migrating database Character set, the figure prove the availability of the Arabic data while the Database Charset is WE8MSWIN1252 (Fig. 13 and 14).

At this step the operation must simulate the extraction process which will be done internally by the Charcon algorithm using database metadata, here there is a need to use third party tool which is aqua data studio or Oracle SQL developer to extract the table's data to flat UTF file, this step will keep the data in temporary oracle managed file, the file will be organized by the (MAS) module triggered by Charcon in simple indexed way in case of multiple schemas, the index will use schema name plus type of the data inside file, for example, if schema is x then the file which holds the data will be, x.data and for the file which holds the triggers will be triggered and so on until all objects are extracted and the file itself will be limited to 1000 rows only and another file will follow the first one for easy import later. Now, metadata SQL code is going to extract the data using Oracle metadata package.

Figure 14 shows the content of the temporary flat UTF file created to hold the Charcon schema data, the sample file holds one row contain Arabic data and the flat file can be viewed normally using any text editor in case of small data or any special editor like Ultra Edit for huge data.

Now after extracting the data index by schema name in flat files, now real table in the database must be emptied to simulate step 3 in Fig. 11, notice that tables extracted in the previous step should be empty which will be automatically done by MAS module, this must do manually as shown in Fig. 15.

After removing all Arabic data from the schemas we need to change the database character set manually using the following list of commands:

• ALTER SYSTEM ENABLE RESTRICTED SESSION; ALTER SYSTEM SET JOB_QUEUE_PROCESSES=0; ALTER SYSTEM SET AQ_TM_PROCESSES=0; ALTER DATABASE OPEN
• ALTER DATABASE CHARACTER SET INTERNAL_USE AR8MSWIN1256; ALTER SYSTEM DISABLE RESTRICTED SESSION

```
SQL> conn charcon/c;
Connected.
SQL> select * from val;

NAME                                                   AREA
-------------------------------------------------   -----------------
مجتبى الطيب                                            port sudan

SQL> delete val;

1 row deleted.

SQL> commit;

Commit complete.

SQL> |
```

Fig. 15: Data deletion from schemas

```
Total System Global Area   135338868 bytes
Fixed Size                     453492 bytes
Variable Size              109051904 bytes
Database Buffers            25165824 bytes
Redo Buffers                  667648 bytes
Database mounted.
Database opened.
SQL> alter system enable restricted session;

System altered.

SQL> alter database character set INTERNAL_USE AR8MSWIN1256;

Database altered.

SQL> alter system disable restricted session;

System altered.

SQL>
```

Fig. 16: Character set conversion

The database now is AR8MSWIN1256 which is different from the source database character set, then it starts reading the flat files and import the contents, here the MAS will take filenames as the Schema and the file extension as the type of data will be imported to do that manually the procedure must read every file alone with SQL command as in Fig. 16.

At this point the Charcon is responsible to Alert the DBA to change the character set for Window registry to fit the target character set and confirm the import, this step must be done manually and carefully because the tool will be used to read the flat files which contain the rows of data is the SQL plus which is acting as a client program for the Oracle database, so, the registry value must be changed to the appropriate one in this case, It changed it to match the target character set which is AR8MSWIN1256, after changing the registry as requested SQL Plus started to read the flat files and import the data manually as shown in Fig. 17.

After importing all the flat files and enabling the constraints in the tables the Arabic data will represent correctly in the new character set as if the nothing happens to the database, no damage and no even partial corruption (Fig. 18) shows the same old data in the Charcon schema after migrating the database character set to the AR8MSWIN1256 which will cause damage to the Arabic data inside in normal case.

Execution result: The goal of Charcon is to protect Arabic data within Oracle database against corruption, which is caused by character set conversion to non-superset character and to enable a new method to move
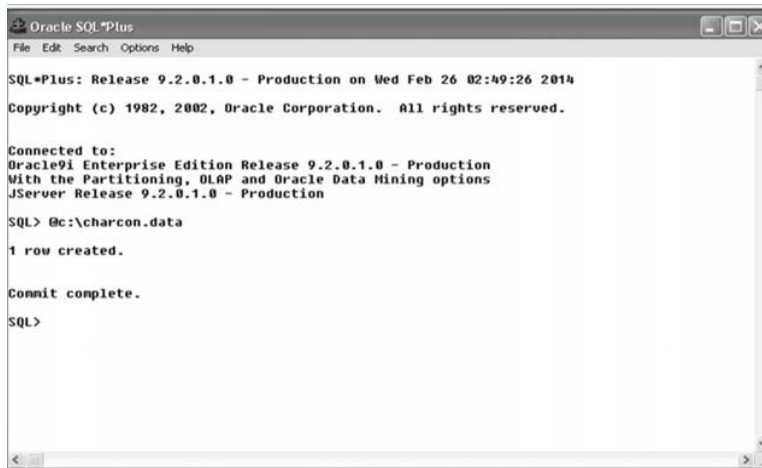
Fig. 17: Data imported from flat file

Fig. 18: Arabic data represented correctly after conversion

Arabic data faster between character sets due to the need of application upgrade or environment changes, the Charcon is successful and reaches its goal.

CONCLUSION

Most of application and database application developers suffer while their development life cycle from supporting the Arabic language, since, it has different writing direction and different way of way to draw letters than the standard language which is English.

Thus, the Arabic language is always the less supported language in application and software platform, so this thesis is basically discussed one of the Arabic related problems in Oracle database. Since, Oracle database supported a large collection of languages and it supports most operation including character set migration operation which causes corruption to Arabic data that were stored in the database and not the English data when migrating to the non-superset character set. The database mechanism is unsuccessful with Arabic data Migration to another character set that uses the same byte length 1 byte as the source, the migration process was successful with English Letters but the Arabic letter is damaged.

oracle advice to use UTF-8 character set as target migration but there are many reasons not to use this character set because it stores the letter is 2 bytes while the Arabic letter require 1 byte, so the consumption of storage and the performance issues in addition to the incompatibility with some application, all these lead to design of character set converter Charcon which allows Arabic data migration from character set to non-superset

character set, it also protects the migrated data in case of corruption, it also supports the reverse migration from superset to subset which is not available in Arabic database. Charcon is able to migrate other languages more over Arabic as long as the target character set Supports the language is going to upgrade.

Charcon designed to be interactive, it requests it's running parameters and the final approval to execute the operation from the Database administrator. This operation may speed down the Charcon module performance but it allows the module to protect the data against corruption. Charcon was designed to work as a part of Oracle Software within the database itself to work fast and efficiently but it can be stand alone as third party module but it's startup operation will be manually by the database Administrator.

Charcon module simulation procedures were tested in real Oracle database uses WE8MSWIN1252 character set and with Charcon it migrated to AR8MSWIN1256, the process was manual because the Module is not implemented as a part of the database, the operation was very successful.

RECOMMENDATIONS

The goal of Charcon is to protect Arabic data with in Oracle database against corruption which caused by Character set conversion to non-superset character in addition to enable a new powerful feature to give the database the ability to move Arabic data faster between character sets, the Charcon is successful and reach its goal, In simulation and Manual implementation of Charcon module in the real life some issues may occur, these issues didn't affect the overall function of Charcon but may slow down the entire process in some circumstance, these issues are mentioned.

Performance issue with big databases: With huge databases like Gigabytes of data the Charcon module may face some problems in injecting the data rows to the Flat files, the Charcon Conversion Mechanism is FFD (Flat File Dependable) which means that at every step of executing the dump and restore procedure to protect the Arabic data the process will depend in the Flat file, this slow down the process since every object in the database that needs to be protected will issue Read, Write operation to the file and the more number of database objects the more time the Charcon will spend protecting data, this will maximize the time of doing the job, the plan of using (MPPE) Multi Process Parallel Execution in Charcon will minimize this time, the more parallel process the less time.

Multi-process parallel execution resource consumption: To minimize the processes time of Charcon it requires to enable MPPE which is Multi Process Parallel execution, it is a way of allowing multi instance of Charcon Algorithm to work at the Same time sharing the same job, let's assume that a 2 Gigabyte Database with 20000 Objects may need 15 min to Migrated safely using Charcon to Target Character set, when two instance allowed the time will decrease to 7 min and for Four Instances the process will take 3.5 min only, this will greatly optimize the time needed to migrate the data but will consume more resource at the same time, let's assume the single instance of Charcon will require 50 mb of Machine RAM and 5.0% of processor to run four Instances it requires 200 mb of RAM and 20.0% of processor power.

Incomplete conversion recovery process interruption: Database recovery is highly and most critical feature must be available when executing any process in the database while Charcon works inside the database and touches the data then a recovery scenario must be provided in case of any interruption happen to the conversion process, the interruption may lead to data loss to one or more Schema, there, Charcon will not touches the data or made any changes to the real data until the hall preparation process Completed, then it start empting data location for Conversion process, this procedure will done after taking backup of the affected data and if any failure occurs at this step Charcon will restore all data to its original location while restoring if any another failure occur like power outage or disk failure the process will be un recoverable, this will lead to data loss, the future enhancement of Charcon should carefully take this issue in consideration.

REFERENCES

01. Markus, M.L. and C. Tanis, 2000. The Enterprise Systems Experience-from Adoption to Success. In: Framing the Domains of IT Research: Glimpsing the Future Through the Past, Zmud, R.W. (Ed.). Pinnaflex Educational Resources, Cincinnati, OH, USA., pp: 173-207.
02. Elmasri, R. and S.B. Navathe, 2007. Fundamentals of Database Systems. 5th Edn., Addison-Wesley, USA., ISBN-13: 9780321369574, Pages: 1139.
03. Loney, K., 2004. Oracle Database 10g: The Complete Reference. 1st Edn., McGraw-Hill Book Co., New York, ISBN-13: 978-0072253511, pp: 1200.
04. Coronel, C. and M. Steven, 2016. Database Systems: Design, Implementation and Management. Cengage Learning, Boston, Massachusetts, USA.,.
05. Antonioletti, M., M. Atkinson, R. Baxter, A. Borley and N.P.C. Hong *et al*., 2005. The design and implementation of grid database services in OGSA-DAI. Concurrency Comput.: Pract. Experience, 17: 357-376.
06. Paul, L., 2004. Oracle Database Globalization Support Guide10g Release 1. 1st Edn., Oracle Press, California, USA.,.

07. Veditz, D.P. and D.B. Schnepper, 2003. System and method for national language support. U.S. Patent No. 6,507,813, U.S. Patent and Trademark Office, Washington, USA.

08. Chiba, D., W. Chu, C. Ho, G. Hua and S. Law *et al.*, 2005. Oracle Database Globalization Support Guide, 10g Release 2 (10.2) B14225-02. Oracle Press, California, USA.,.

09. Arunagirinathan, S., V. Deep, A. Kumar, E. Lee and S. Marya *et al.*, 2003. Oracle application server portal error messages guide, 10g (9.0.4) Part no. B10608-01. Oracle Corporation, California, USA.

10. Schwartz, B., P. Zaitsev and V. Tkachenko, 2012. High Performance MySQL: Optimization, Backups and Replication. O'Reilly Media, Cambridge, England, Pages: 771.