

A Tinkerer's Guide to Structured Query Language in Déjà Vu X

© Michael Farrell 2004-2005

Table of Contents

Table of Contents	1
Preamble, Disclaimer and Important Warning	3
Important Warning.....	3
Disclaimer	3
Copyright.....	3
The Author	3
Price	3
0) Introduction.....	4
What is Structured Query Language?	4
What has Structured Query Language got to do with Déjà Vu X?.....	4
What is a relational database?	4
What can I do with Structured Query Language in Déjà Vu X?.....	4
Where do I enter SQL Select Statements and SQL Commands?	4
1) Structured Query Language in the Project File.....	5
Database structure	5
SQL select statements	5
Example select statements.....	5
Delete commands	7
Update commands	8
Example update commands	8

2) Structured Query Language in the Memory Database.....	10
Database structure	10
Relationship between main database tables	10
SQL select statements	10
Example select statements for main tables	10
Delete commands	14
Example delete commands for the Translations table	14
Example delete commands for the Sentences table (delete defined and all associated sentences)	15
Example delete commands for the Sentences table (only delete defined sentences)	16
Update commands	17
Example update commands for main tables	17
3) Structured Query Language in the Terminology Database.....	19
Database structure	19
Relationship between main database tables	19
SQL select statements	19
Example select statements for Lemmas and Relations tables.....	19
Example select statements for Attributes table	22
Meanings of Category IDs (Categories Table) in ATRIL Déjà Vu X format TDBs.....	22
Delete commands	23
Example delete commands for Lemmas and Relations tables	23
Example delete commands for Attributes table	26
Update commands	28
Example update commands for Lemmas and Relations tables	28
Example update commands for Attributes table	30
Insert commands.....	32
Example insert commands for Attributes table	32

Preamble, Disclaimer and Important Warning

Important Warning

Reckless use of Structured Query Language in Déjà Vu X may permanently damage your databases. To prevent serious irreversible consequences you are strongly advised to **make back-up copies of your files before executing any SQL commands**.

Disclaimer

This document is provided "as is" and its use is entirely at the user's own risk. In no event will the Author be liable for any damages, including but not limited to any lost profits, lost savings or other incidental or consequential damages arising out of the use or the inability to use this document, even if the Author has been advised of the possibility of such damages.

Copyright

This document is copyrighted by its Author: © Michael Farrell 2004-2005. All rights reserved. Other brand and product names are trademarks or registered trademarks of their respective holders. This document may be distributed freely so long as it is unaltered.

The Author

If you are overcome by an irresistible urge to find out more about the person behind this document, you can read something here:
<http://www.traduzioni-inglese.it>

Price

This document is provided free of charge. If you feel guilty about using this document for free and think you really should have paid for it, you might like to know that the Author would be very happy if you made a donation to your local branch of UNICEF.

0) Introduction

This document is not designed to replace the equivalent section of the official Déjà Vu X manual, but to supplement it. You are therefore advised to consult the manual along with this document, since it contains important information not contained herein. Version 1.5 of this document refers to Déjà Vu X version 7.0.273. Since ATRIL is constantly enhancing the program, some of the information in this document may not be valid for earlier or later builds.

What is Structured Query Language?

Structured Query Language (SQL) is a powerful language for manipulating databases.

What has Structured Query Language got to do with Déjà Vu X?

The Memory File, Terminology Database and Project File are all relational databases, which may be opened with MS Access 2000 or better (right click the file in Windows Explorer, choose "Open with...", then Microsoft Access). They can also be manipulated in Access, although ATRIL strongly advises you not to do so. Above all, however, they can be manipulated using SQL.

What is a relational database?

A relational database is a database where the fields (single pieces of information, e.g. Client, Subject, etc.) and records (sets of fields, e.g. sentence along with its corresponding language code and the date on which it was entered, etc.) are organized into several tables (in which the rows are the records and the columns, the fields) linked to each other by special key fields (IDs).

What can I do with Structured Query Language in Déjà Vu X?

Lots of things. For example you can correct a miss-associated client or subject in a Memory File or Terminology Database, or export almost any subset of those files on the basis of the most unimaginable combination of conditions (e.g. everything apart from sentences translated by Tom on the morning of 1 January 2000 when his hangover severely compromised his work). For a real idea of what you can do, see the examples in the various sections of this document and in the Déjà Vu X manual.

Where do I enter SQL Select Statements and SQL Commands?

In the Project, Memory Database and Terminology Database interfaces and in the Memory Database and Terminology Database Export Wizard Filtering Conditions windows. See Déjà Vu X manual for details.

1) Structured Query Language in the Project File

Database structure

All the main fields are found on the **Pairs** table

SQL select statements

To enter **Select statements** in the Project interface, click the "Rows Selector" (the second record selector in the selector row – see Déjà Vu X manual) and choose "SQL Statement" from the dropdown box.

Some of the example statements below are equivalent to items in the Rows Selector dropdown box itself, but they may be combined together to display records which cannot be displayed in any other way (see notes after chart).

Example select statements

Field	SQL statement	Notes
Bookmark_2057	Bookmark_2057 = True	Display all sentence pairs whose language 2057 target segment is marked with a bookmark.
Comments	Comments = 'Careful with this sentence!'	Display all sentence pairs whose source segment is associated with the comment "Careful with this sentence!".
Comments_2057	Comments_2057 = 'Brilliant translation!'	Display all sentence pairs whose language 2057 target segment is associated with the comment "Brilliant translation!".
Duplicate	Duplicate = True	Display all sentence pairs with duplicate source. This is equivalent to choosing "Duplicate Rows" from the Rows Selector dropdown box.
FileID	FileID = 1	Display all sentence pairs in file ID No. 1. This is equivalent to choosing the appropriate file from the File Navigator. It only works if the whole project is displayed as if it were a single file. Although the Lexicon is assigned FileID = -1, this Select statement cannot be used to display it. However this fact may be useful to bear in mind when writing commands.
Freq	Freq > 30	Display all lexicon entries which occur more than 30 times. It only works when the Lexicon is displayed.
LastModified_2057	LastModified_2057 >= #02/10/2004#	Display all sentence pairs whose language 2057 target segment was modified on or after 10 February 2004.
LexWords	LexWords = 2	Display all lexicon entries consisting of 2 words. It only works when the Lexicon is displayed.
Percentage_2057	Percentage_2057 >= 95	Display all segments whose language 2057 target segment is a greater than or equal to 95 percent match.
Source	Source LIKE '*tele*'	Display all sentence pairs whose source segment contains the letters "tele" together in that order.

Field	SQL statement	Notes
Status_2057	<code>((Status_2057 \ 1) MOD 2) = 1) AND (((Status_2057 \ 512) MOD 2) = 1)</code>	Display all sentence pairs whose language 2057 target segment is a locked exact match. Use "Build Expression" for this type of query.
Target_2057	<code>Target_2057 LIKE 'C'</code>	Display all sentence pairs whose language 2057 target segment starts with the letter "C".
Terms_2057	<code>Terms_2057 <>"</code>	Display all sentence pairs whose language 2057 target segment has been marked as containing inconsistent terminology. This is equivalent to choosing "Rows with Inconsistent Terminology" from the Rows Selector dropdown box.
User_2057	<code>User_2057 = 'Tom'</code>	Display all sentence pairs whose language 2057 target segment was last modified by "Tom".

You can combine two or more conditions together with AND or OR, like this:

`FileID = 1 AND User_2057 = 'Tom'`

You can also make "everything except" statements using 'NOT':

`NOT User_2057 = 'Tom'` (which is the same as `User_2057 <> 'Tom'`)

For a full list of wildcards which may be used with the LIKE operator, see "lookup with wildcards" in the DVX on-line help or manual.

For a list of language codes used in field names (`Bookmark_xxxx`, `Comments_xxxx`, `LastModified_xxxx`, `Percentage_xxxx`, `Status_xxxx`, `Target_xxxx`, `Terms_xxxx` and `User_xxxx`), see "List of Language and Sublanguage Codes" in the DVX on-line help or manual.

Various Visual Basic functions may be used in Déjà Vu X SQL expressions, for example:

`LastModified_2057 >= Date()`

displays all sentence pairs whose language 2057 segments have been modified today (assuming the computer time/date setting is correct).

For a full list of Visual Basic functions see the Microsoft Access on-line help.

Delete commands

To enter Delete commands in the project interface, choose "Execute SQL..." from the "Project" menu.

You are strongly advised **not to delete** any part of the project since it would almost certainly make finished file exporting impossible.

If you decide to ignore this advice, the general form of a Delete command is:

```
DELETE FROM Pairs WHERE Select statement
```

where **Select statement** can be any of the SQL statements (or combination thereof) shown on the **Example select statements** chart above.

It is advisable to make sure the **Select statement** gives the expected result by trying it alone in the Project interface first (where possible).

To see the effects of a command you must refresh the display by switching the rows selector from "All Rows" to something else, then back to "All Rows" again.

It is also advisable to repair and compact the database after deleting data to take full advantage of space savings, improve efficiency and make sure the changes take effect ("Tools" menu).

Update commands

To enter Update commands in the project interface, choose "Execute SQL..." from the "Project" menu. To see the effects of a command you must refresh the display by switching the rows selector from "All Rows" to something else, then back to "All Rows" again.

It is advisable to repair and compact the database after updating data to improve efficiency and make sure the changes take effect ("Tools" menu).

Example update commands

Field	SQL statement	Notes
Bookmark_2057	UPDATE Pairs SET Bookmark_2057 = True WHERE Select statement	Mark all sentence pairs in language 2057 identified by the Select statement with a bookmark.
Comments	UPDATE Pairs SET Comments = "Careful with this sentence!" WHERE Select statement	Associate the comment "Careful with this sentence!" with the source segments of all sentence pairs identified by the Select statement .
Comments_2057	UPDATE Pairs SET Comments_2057 = "Brilliant translation!" WHERE Select statement	Associate the comment "Brilliant translation!" with the language 2057 target segments of all sentence pairs identified by the Select statement .
Duplicate	UPDATE Pairs SET Duplicate = False WHERE Select statement	Mark all sentence pairs identified by the Select statement as not having duplicate source segments. It is hard to imagine when this might be useful.
FileID	UPDATE Pairs SET FileID = 1 WHERE Select statement	Associate the sentence pairs identified by the Select statement with File ID No. 1. This would almost certainly make finished file exporting impossible.
Freq	UPDATE Pairs SET Freq = 30 WHERE Select statement	Mark all lexicon entries identified by the Select statement as occurring 30 times.
LastModified_2057	UPDATE Pairs SET LastModified_2057 = #02/10/2004# WHERE Select statement	Mark the language 2057 target segments of all sentence pairs identified by the Select statement as having been last modified on 10 February 2004.
LexWords	UPDATE Pairs SET LexWords = 2 WHERE Select statement	Mark all lexicon entries identified by the Select statement as consisting of 2 words. It is hard to imagine when this might be useful.
Percentage_2057	UPDATE Pairs SET Percentage_2057 = 95 WHERE Select statement	Mark the language 2057 target segments of all sentence pairs identified by the Select statement as having a score of 95 percent. It does not change the status of the segment, i.e. an assembled row stays such.
Source	UPDATE Pairs SET Source = 'Bad paragraph' WHERE Select statement	Change the source text of all sentence pairs identified by the Select statement into 'Bad paragraph'. If the original source text contained embedded codes, this might make finished file exporting impossible.
Status_2057	UPDATE Pairs SET Status_2057 = Status_2057 + 2048 WHERE Select statement	Prevent all sentence pairs identified by the Select statement from being sent to the memory database.

Field	SQL statement	Notes
Target_2057	UPDATE Pairs SET Target_2057 = 'Retranslate this segment' WHERE Select statement	Change the language 2057 target segments of all sentence pairs identified by the Select statement into 'Retranslate this segment'.
Terms_2057	UPDATE Pairs SET Terms_2057 = " WHERE Select statement	Mark the language 2057 target segments of all sentence pairs identified by the Select statement as not containing inconsistent terminology
User_2057	UPDATE Pairs SET User_2057 = 'Tom' WHERE Select statement	Mark the language 2057 target segments of all sentence pairs identified by the Select statement as having been last modified by 'Tom'.

The [Select statement](#) may be any of the statements (or combination thereof) shown on the **Example select statements** chart. It is advisable to make sure the [Select statement](#) gives the expected result by trying it alone in the Project interface first (where possible).

The field in the [Select statement](#) can also be the same as the field which needs updating, for example:

```
UPDATE Pairs SET User_2057 = 'Tom' WHERE User_2057 = 'Dick'
```

corrects the translator's name from Dick to Tom.

You can update several fields at once in this way:

```
UPDATE Pairs SET User_2057 = 'Tom', Comments_2057 = "Brilliant translation!" WHERE User_2057 = 'Dick'
```

For a list of language codes used in field names ([Bookmark_xxxx](#), [Comments_xxxx](#), [LastModified_xxxx](#), [Percentage_xxxx](#), [Status_xxxx](#), [Target_xxxx](#), [Terms_xxxx](#) and [User_xxxx](#)) see "List of Language and Sublanguage Codes" in the DVX on-line help or manual.

Various Visual Basic functions may be used in Déjà Vu X SQL commands, for example:

```
UPDATE Pairs SET Target_2057 = RTrim (Target_2057)
```

removes all spaces from the end of target segments. For a full list of Visual Basic functions see the Microsoft Access on-line help.

2) Structured Query Language in the Memory Database

Database structure

Relationship between main database tables

Translations	<->	Sentences
ID		ID
Client		DateStamp
FileID		Duplicate
PrjID		IsSource
RowID		Lang
Subject		Sentence
Export Wizard SQL Select statements refer directly to this table		SubLang
		UserNick
		Memory Database interface and Export Wizard SQL Select statements refer directly to this table

SQL select statements

To enter [Select statements](#) in the Memory Database interface, click "SQL..." in the upper middle part of the interface itself (immediately above the source segment column, on the right – see Déjà Vu X manual). The same [Select statements](#) may be entered during the Export Wizard on the Filtering Conditions window (File>Export>File or Database...).

Example select statements for main tables

Field	Table	SQL statement	Notes
Client	Translations	ID IN (SELECT ID FROM Translations WHERE Client = '01')	Export/Display all sentence pairs entered for Client No. 01.
DateStamp	Sentences	ID IN (SELECT ID FROM Sentences WHERE DateStamp >= #02/10/2004#)	Export/Display all sentence pairs where one or both sentences in the pair was entered on or after 10 February 2004.
Duplicate	Sentences	ID IN (SELECT ID FROM Sentences WHERE Duplicate = True)	Export all sentence pairs where one or both sentences in the pair is marked as a duplicate. As regards display, this is equivalent to choosing "Duplicate Only" on the Memory Database interface.
FileID	Translations	ID IN (SELECT ID FROM Translations WHERE FileID = 2 AND PrjID = 7055476)	Export/Display all sentence pairs added from file No. 2 of project No. 7055476.

Field	Table	SQL statement	Notes
IsSource	Sentences	ID IN (SELECT ID FROM Sentences WHERE IsSource = False)	Export/Display all sentence pairs in which one of the two language segments was not entered as a source segment. In other words it is equivalent to exporting/displaying everything. The short form may be of more practical use (see note below).
Lang	Sentences	ID IN (SELECT ID FROM Sentences WHERE Lang = 9)	In theory export/display all sentence pairs in which one of the two language segments is in English (language = 9), but in practice the appropriate choices are made on the Memory Database interface/Export wizard.
PrjID	Translations	ID IN (SELECT ID FROM Translations WHERE PrjID = 7055476)	Export/Display all sentence pairs added from project No. 7055476
RowID	Translations	ID IN (SELECT ID FROM Translations WHERE RowID = '0000052' AND PrjID = 7055476)	Export/Display the sentence pairs added from Row No. 0000052 of project No. 7055476
Sentence	Sentences	ID IN (SELECT ID FROM Sentences WHERE Sentence LIKE '*pho*')	Export/Display all sentence pairs where one or both sentences in the pair contain the letters "pho" together in that order
Subject	Translations	ID IN (SELECT ID FROM Translations WHERE Subject = '0')	Export/Display all sentence pairs associated with subject No. 0
SubLang	Sentences	ID IN (SELECT ID FROM Sentences WHERE SubLang = 2 AND Lang = 9)	In theory export/display all sentence pairs in which one of the two language segments is in British English (language = 9, sublanguage = 2), but in practice the appropriate choices are made on the Memory Database interface/Export wizard.
UserNick	Sentences	ID IN (SELECT ID FROM Sentences WHERE UserNick = 'Tom')	Export/Display all sentence pairs where one or both sentences in the pair was entered by Tom.

In reality the **Select statements** shown above are ANDed with the language chosen as Source Language on the Memory Database interface or for the Main Sentence in the Export Wizard, i.e. the example shown for **UserNick** above actually corresponds to:

```
ID IN (SELECT ID FROM Sentences WHERE UserNick = 'Tom') AND SubLang = 2 AND Lang = 9
```

(assuming British English is chosen).

Select statements may be shortened (i.e. only the part after WHERE may be used) in certain circumstances. All statements may be shortened in the Export Wizard, but only the statements relating to the **Sentences** table may be shortened on the Memory Database interface. For example:

```
Client = '01'
```

may be used to export sentence pairs corresponding to Client No. 01 and

`DateStamp >= #02/10/2004#`

may be used both during the Export Wizard and in the Memory Database interface to select sentences according to their date of entry. There is however a small difference in the way `Sentences` table long and short forms work in the Memory Database interface: the long form works on the whole database, whereas the short form only works on the sentences displayed in the source (left-hand) column. In other words:

`ID IN (SELECT ID FROM Sentences WHERE Sentence LIKE '*pho*')`

displays all sentence pairs where **one or both** sentences in the pair (including any paired sentences in languages other than those displayed) contain the letters "pho" together in that order, whereas

`Sentence LIKE '*pho*'`

only displays sentence pairs where the **sentence in the source column** contains the letters "pho" together in that order. In the Export Wizard, `Sentences` table short forms have the same effect as the long forms would have.

You can combine two or more conditions together for fields **from the same table** with AND or OR, like this:

`ID IN (SELECT ID FROM Translations WHERE Client = '01' AND Subject = '0')`
(or, in the Export Wizard only, `Client = '01' AND Subject = '0'`)

You can combine two or more conditions together for fields **from different tables** with AND or OR, like this:

`ID IN (SELECT ID FROM Translations WHERE Client = '01') AND ID IN (SELECT ID FROM Sentences WHERE DateStamp >= #02/10/2004#)`

You can also make "everything except" statements using 'NOT':

`ID IN (SELECT ID FROM Translations WHERE NOT Client = '01')`
which is the same as `ID IN (SELECT ID FROM Translations WHERE Client <> '01')`

For a full list of wildcards which may be used with the LIKE operator, see "lookup with wildcards" in the DVX on-line help or manual.

For a list of language and sublanguage codes which may be used in the **Lang** and **SubLang** fields, see "List of Language and Sublanguage Codes" in the DVX on-line help or manual.

Various Visual Basic functions may be used in Déjà Vu X SQL expressions, for example:

```
ID IN (SELECT ID FROM Sentences WHERE DateStamp >= Date())
```

exports/displays all sentence pairs where one or both sentences in the pair was entered today (assuming the computer time/date setting is correct). For a full list of Visual Basic functions see the Microsoft Access on-line help.

Delete commands

To enter Delete commands in the Memory Database interface, choose "Execute SQL..." from the "Database" menu. After deleting data from the **Sentences** table, the display must be refreshed by closing and reopening the database. It is advisable to repair and compact the database after deleting data to take full advantage of space savings, improve efficiency and make sure the changes take effect.

Example delete commands for the **Translations** table

Field	SQL statements	Notes
Client	1) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Translations WHERE Client = '01') 2) DELETE FROM Translations WHERE Client = '01'	Delete all sentences entered for Client No. 01.
FileID	1) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Translations WHERE FileID = 2 AND PrjID = 7055476) 2) DELETE FROM Translations WHERE FileID = 2 AND PrjID = 7055476	Delete all sentences entered from file No. 2 of project No. 7055476.
PrjID	1) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Translations WHERE PrjID =1234567) 2) DELETE FROM Translations WHERE PrjID =1234567	Delete project 1234567 from memory database.
RowID	1) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Translations WHERE RowID = '0000052' AND PrjID = 7055476) 2) DELETE FROM Translations WHERE RowID = '0000052' AND PrjID = 7055476	Delete sentence pair No. 52, project No. 7055476.
Subject	1) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Translations WHERE Subject = '0') 2) DELETE FROM Translations WHERE Subject = '0'	Delete all sentences entered for Subject No. 0

Since the data relating to the sentences is spread across two different tables (**Translations** and **Sentences**), it must be deleted from both tables, which means using two separate **Delete statements**. The order of the two operations is extremely important: the data in the table containing the reference field must be deleted last. For example, to delete project 1234567 from the memory database the commands are:

- 1) DELETE FROM **Sentences** WHERE ID IN (SELECT ID FROM **Translations** WHERE **PrjID**=1234567)
- 2) DELETE FROM **Translations** WHERE **PrjID**=1234567

Statement 1 deletes the records on the **Sentences** table which relate to the records on the **Translations** table entered from Project ID number 1234567 and statement 2 deletes the records on the **Translations** table entered from Project ID number 1234567. If the Delete commands are entered the other way round, the first statement would delete the records on the **Translations** table entered from

Project ID number 1234567, but the second command would do nothing, since it would no longer be possible to establish which records on the **Sentences** table had related to the now deleted records on the **Translations** table.

If sentences have been added manually or "consolidated" on the Memory Database interface, the **Sentences** table attributes of paired sentences may not correspond, i.e. they may be marked as entered by different users on different dates. Similarly the **Lang**, **SubLang**, **IsSource**, **Duplicate** and **Sentence** fields are never (or may not be) the same for both halves of the pair. Consequently when you enter a Delete command based on **Sentences** table fields you have to decide if you want to delete only one half of the pair or a sentence with all its associated sentences (the other pair halves in all the other languages). Decide what you really want to do then choose from the appropriate **Delete command table** below.

Example delete commands for the **Sentences table (delete defined and all associated sentences)**

Field	SQL statements	Notes
DateStamp	1) DELETE FROM Translations WHERE ID IN (SELECT ID FROM Sentences WHERE DateStamp >= #02/10/2004#) 2) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Sentences WHERE DateStamp >= #02/10/2004#)	Delete all sentences entered on or after 10 February 2004 and all sentences paired with them.
Duplicate	1) DELETE FROM Translations WHERE ID IN (SELECT ID FROM Sentences WHERE Duplicate = True) 2) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Sentences WHERE Duplicate = True)	Delete all sentences marked as duplicates and all sentences paired with them.
IsSource	1) DELETE FROM Translations WHERE ID IN (SELECT ID FROM Sentences WHERE IsSource = False) 2) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Sentences WHERE IsSource = False)	Delete all sentences which were originally entered as target segments and all sentences paired with them. In theory this is equivalent to deleting all sentences.
Lang	1) DELETE FROM Translations WHERE ID IN (SELECT ID FROM Sentences WHERE Lang = 9) 2) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Sentences WHERE Lang = 9)	Delete all sentences in English (language No. 9) and all sentences paired with them.
Sentence	1) DELETE FROM Translations WHERE ID IN (SELECT ID FROM Sentences WHERE Sentence LIKE "pho*") 2) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Sentences WHERE Sentence LIKE "pho*")	Delete all sentences which contain the letters "pho" together in that order and all sentences paired with them.
SubLang	1) DELETE FROM Translations WHERE ID IN (SELECT ID FROM Sentences WHERE SubLang = 2 AND Lang = 9) 2) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Sentences WHERE SubLang = 2 AND Lang = 9)	Delete all sentences in British English (language No. 9, sublanguage No. 2) and all sentences paired with them.

Field	SQL statements	Notes
UserNick	1) DELETE FROM Translations WHERE ID IN (SELECT ID FROM Sentences WHERE UserNick ='Tom') 2) DELETE FROM Sentences WHERE ID IN (SELECT ID FROM Sentences WHERE UserNick ='Tom')	Delete all sentences entered by "Tom" and all sentences paired with them.

Example delete commands for the **Sentences table (only delete defined sentences)**

Field	SQL statements	Notes
DateStamp	DELETE FROM Sentences WHERE DateStamp >= #02/10/2004#	Delete all sentences entered on or after 10 February 2004.
Duplicate	DELETE FROM Sentences WHERE Duplicate = True	Delete all sentences marked as duplicates.
IsSource	DELETE FROM Sentences WHERE IsSource = False	Delete all sentences which were originally entered as target segments.
Lang	DELETE FROM Sentences WHERE Lang = 9	Delete all sentences in English (language No. 9).
Sentence	DELETE FROM Sentences WHERE Sentence LIKE '*pho*'	Delete all sentences which contain the letters "pho" together in that order.
SubLang	DELETE FROM Sentences WHERE SubLang = 2 AND Lang = 9	Delete all sentences in British English (language No. 9, sublanguage No. 2)
UserNick	DELETE FROM Sentences WHERE UserNick ='Tom'	Delete all sentences entered by "Tom"

For a full list of wildcards which may be used with the LIKE operator, see "lookup with wildcards" in the DVX on-line help or manual.

For a list of language and sublanguage codes which may be used in the **Lang** and **SubLang** fields, see "List of Language and Sublanguage Codes" in the DVX on-line help or manual.

Various Visual Basic functions may be used in Déjà Vu X SQL expressions, for example:

```
DELETE FROM Sentences WHERE DateStamp >= Date()
```

deletes all sentences entered today (assuming the computer time/date setting is correct). For a full list of Visual Basic functions see the Microsoft Access on-line help.

Update commands

To enter Update commands in the Memory Database interface, choose "Execute SQL..." from the "Database" menu. To see the effects of a command you must refresh the display by closing and reopening the database. It is advisable to repair and compact the database after updating data to improve efficiency and make sure the changes take effect ("Tools" menu).

Example update commands for main tables

Field	Table	SQL statement	Notes
Client	Translations	UPDATE Translations SET Client = '01' WHERE Select statement	Set the client associated with all sentences identified by the Select statement to No. 01
DateStamp	Sentences	UPDATE Sentences SET DateStamp = #02/10/2004# WHERE Select statement	Set the entry date associated with all sentences identified by the Select statement to 10 February 2004
Duplicate	Sentences	UPDATE Sentences SET Duplicate = True WHERE Select statement	Mark all sentences identified by the Select statement as duplicates
FileID	Translations	UPDATE Translations SET FileID = 2 WHERE Select statement	Mark all sentences identified by the Select statement as having been entered from file No. 2 of their respective projects.
IsSource	Sentences	UPDATE Sentences SET IsSource = True WHERE Select statement	Mark all sentences identified by the Select statement as being source sentences
Lang	Sentences	UPDATE Sentences SET Lang = 9 WHERE Select statement	Mark all sentences identified by the Select statement as being in English (language No. 9)
PrjID	Translations	UPDATE Translations SET PrjID=1234567 WHERE Select statement	Mark all sentences identified by the Select statement as having been entered from project No. 1234567
RowID	Translations	UPDATE Translations SET RowID = '0000052' WHERE Select statement	Mark all sentences identified by the Select statement as having been entered from row No. 52 of their respective projects
Sentence	Sentences	UPDATE Sentences SET Sentence = 'Surprise!' WHERE Select statement	Change all sentences identified by the Select statement to "Surprise!"
Subject	Translations	UPDATE Translations SET Subject = '0' WHERE Select statement	Set the subject associated with all sentences identified by the Select statement to No. 0
SubLang	Sentences	UPDATE Sentences SET SubLang = 2 WHERE Select statement	Mark all sentences identified by the Select statement as being in sublanguage No. 2 of their respective languages
UserNick	Sentences	UPDATE Sentences SET UserNick = 'Tom' WHERE Select statement	Mark all sentences identified by the Select statement as having been entered by "Tom"

The [Select statement](#) may be any of the statements (or combination thereof) shown on the **Example select statements** chart, but the short form can only be used if the field in the [Select statement](#) is on the same table as the field to be updated, for example:

```
UPDATE Translations SET Client = '01' WHERE Subject = '0'
```

(Client and Subject are both on the Translations table, therefore the short form may be used)

```
UPDATE Translations SET Client = '01' WHERE ID IN (SELECT ID FROM Sentences WHERE DateStamp >= #02/10/2004#)
```

(Client is on the Translations table, but DateStamp is on the Sentences table, therefore the long form is needed)

It is advisable to make sure the [Select statement](#) gives the expected result by trying it alone in the Memory Database interface first (where possible). Bear in mind however that [Select statements](#) in the Memory Database interface are ANDed with the language chosen as Source Language (see note under **SQL Select statements**).

If no [Select statement](#) is added, i.e.:

```
UPDATE Translations SET Client = '01'
```

the client associated with all the sentence in the Memory Database will be set to client number 01. The field in the [Select statement](#) can also be the same as the field which needs updating, for example:

```
UPDATE Translations SET Client = '01' WHERE Client = '02'
```

corrects the client number from 02 to 01. You can update several fields **on the same table** at once in this way:

```
UPDATE Translations SET Client = '01', Subject = '0' WHERE PrjID=1234567
```

For a list of language and sublanguage codes which may be used in the [Lang](#) and [SubLang](#) fields, see "List of Language and Sublanguage Codes" in the DVX on-line help or manual.

Various Visual Basic functions may be used in Déjà Vu X SQL commands, for example:

```
UPDATE Sentences SET Sentence = RTrim (Sentence)
```

removes all spaces from the end of segments. For a full list of Visual Basic functions see the Microsoft Access on-line help.

3) Structured Query Language in the Terminology Database

Database structure

Relationship between main database tables

Attributes		Lemmas		Relations
LemmaID	<->	ID	<->	DstLemmaID
AttrTypeID		FileID		OrgLemmaID
BoolValue		IsSource		UserNick
CategoryID		Lang		DateStamp
DateValue		Lemma		RelTypeID
IntValue		PrjID		
TextValue		SubLang		
Type		SQL Select statements refer directly to this table		

SQL select statements

To enter [Select statements](#) in the Terminology Database interface, click "SQL..." in the upper middle part of the interface itself (immediately above the source term column, on the right – see Déjà Vu X manual).

The same [Select statements](#) may be entered during the Export Wizard on the Filtering Conditions window (File>Export>File or Database...).

Example select statements for **Lemmas** and **Relations** tables

Field	Table	SQL statement	Notes
DateStamp	Relations	ID IN (SELECT OrgLemmaID FROM Relations WHERE DateStamp >= #02/10/2004#)	Export/Display terms entered or linked on or after 10 February 2004.
FileID	Lemmas	ID IN (SELECT ID FROM Lemmas WHERE FileID = 1 AND PrjID = 5276148)	Export/Display terms entered from file No. 1 of project No. 5276148.
IsSource	Lemmas	ID IN (SELECT ID FROM Lemmas WHERE IsSource = True)	Export/Display all terms entered as source terms.

Field	Table	SQL statement	Notes
Lang	Lemmas	ID IN (SELECT ID FROM Lemmas WHERE Lang = 9)	In theory export/display all English (language = 9) terms, but in practice the appropriate choices are made on the Terminology Database Interface/Export wizard.
Lemma	Lemmas	ID IN (SELECT ID FROM Lemmas WHERE Lemma LIKE "*pho*")	Export/Display all terms containing the letters "pho" together in that order.
PrjID	Lemmas	ID IN (SELECT ID FROM Lemmas WHERE PrjID = 5276148)	Export/Display terms entered from project No. 5276148.
RelTypeID	Relations	ID IN (SELECT OrgLemmaID FROM Relations WHERE RelTypeID = 2)	Export/Display terms entered or linked either as translations (RelTypeID = 1), synonyms (RelTypeID = 2) or antonyms (RelTypeID = 3) in an ATRIL Déjà Vu X format TDB.
SubLang	Lemmas	ID IN (SELECT ID FROM Lemmas WHERE SubLang = 2 AND Lang = 9)	In theory export/display all British English (language = 9, sublanguage = 2) terms, but in practice the appropriate choices are made on the Terminology Database Interface/Export wizard.
UserNick	Relations	ID IN (SELECT OrgLemmaID FROM Relations WHERE UserNick ='Tom')	Export/Display terms entered or linked by Tom.

In reality the **Select statements** shown above are ANDed with the language chosen as Source Language on the Terminology Database interface or for the Main Lemma in the Export Wizard, i.e. the example shown for **UserNick** above actually corresponds to:

ID IN (SELECT OrgLemmaID FROM Relations WHERE UserNick ='Tom') AND SubLang = 2 AND Lang = 9
(assuming British English is chosen).

Select statements can be shortened (i.e. only the part after WHERE may be used) if they relate to **Lemmas** table fields, for example:

PrjID = 5276148

may be used to display/export terms entered from project No. 5276148. If in doubt, use the long statement, which always works.

Since all the relations defined in ATRIL Déjà Vu X format TDBs (Translation, Synonym and Antonym) are symmetric, **OrgLemmaID** may be replaced by **DstLemmaID** in **Select statements** referring to **Relations** table fields without any difference in result, unless asymmetric relations have been introduced through database customization or careless use of Update commands (see note under Update commands). If you have asymmetric relations in your database, you may have to use **DstLemmaID** instead of/combined with (using OR, AND, etc.) **OrgLemmaID**. Note that terms have **Relations** table field values for all the relations (links) they have with other terms. In other words if John entered term A as a translation of term B on 1 February 2004 and Tom "manually" linked the same term A to term C as a synonym on 15 February 2004 (on the Terminology Database Interface) the expressions for **DateStamp**, **RelTypeID**

and **UserNick** shown in the table above will also export/display term A even though it was originally entered at an earlier date by a different user with a different relationship!

You can combine two or more conditions together for fields **from the same table** with AND or OR, like this:

```
ID IN (SELECT ID FROM Lemmas WHERE PrjID = 5276148 AND Lang = 9)
(or, in this case, since the fields are on the Lemmas table, PrjID = 5276148 AND Lang = 9)
```

You can combine two or more conditions together for fields **from different tables** with AND or OR, like this:

```
ID IN (SELECT ID FROM Lemmas WHERE PrjID = 5276148) AND ID IN (SELECT OrgLemmaID FROM Relations WHERE UserNick = 'Tom')
```

You can also make "everything except" statements using 'NOT':

```
ID IN (SELECT ID FROM Lemmas WHERE NOT PrjID = 5276148)
which is the same as ID IN (SELECT ID FROM Lemmas WHERE PrjID <> 5276148)
```

For a full list of wildcards which may be used with the LIKE operator, see "lookup with wildcards" in the DVX on-line help or manual.

For a list of language and sublanguage codes which may be used in the **Lang** and **SubLang** fields, see "List of Language and Sublanguage Codes" in the DVX on-line help or manual.

Various Visual Basic functions may be used in Déjà Vu X SQL expressions, for example:

```
ID IN (SELECT OrgLemmaID FROM Relations WHERE DateStamp >= Date())
```

exports/displays all terms entered or linked with other terms today (assuming the computer time/date setting is correct).

For a full list of Visual Basic functions see the Microsoft Access on-line help.

Example select statements for *Attributes* table

	ATRIL Déjà Vu X TDB	Converted Déjà Vu 3 TDB	Notes
	SQL statement		
Part of Speech	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 1 AND CategoryID = 2)		Export/Display all verbs (see Meanings of Category IDs table below)
Gender	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 2 AND CategoryID = 12)		Export/Display all neutral gender terms (see Meanings of Category IDs table below)
Number	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 3 AND CategoryID = 13)		Export/Display all singular terms (see Meanings of Category IDs table below)
Definition	Field not present	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 4 AND TextValue LIKE "pho*")	Export/Display all terms whose definitions contain the letters "pho" together in that order
Subject	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 4 AND TextValue = '75')	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 5 AND TextValue = '75')	Export/Display terms associated with Subject No. 75
Client	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 5 AND TextValue = '23')	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 6 AND TextValue = '23')	Export/Display terms associated with Client No. 23
Context	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 6 AND TextValue LIKE "pho*")	ID IN (SELECT LemmaID FROM <i>Attributes</i> WHERE AttrTypeID = 7 AND TextValue LIKE "pho*")	Export/Display all terms whose context contains the letters "pho" together in that order

Meanings of Category IDs (Categories Table) in ATRIL Déjà Vu X format TDBs

Part of Speech		Gender		Number	
CategoryID	Value	CategoryID	Value	CategoryID	Value
1	Noun	10	Masculine	13	Singular
2	Verb	11	Feminine	14	Plural
3	Adjective	12	Neutral		
4	Adverb				
5	Article				
6	Preposition				
7	Pronoun				
8	Conjunction				
9	Interjection				

See notes after **Example select statements for *Lemmas* and *Relations* tables** chart.

Delete commands

To enter Delete commands in the Terminology Database interface, choose "Execute SQL..." from the "Database" menu.

After deleting data from the **Lemmas** table, the display must be refreshed by closing and reopening the database.

It is advisable to repair and compact the database after deleting data to take full advantage of space savings, improve efficiency and make sure the changes take effect ("Tools" menu).

Example delete commands for **Lemmas and **Relations** tables**

Field	Table	SQL statements	Notes
DateStamp	Relations	1) DELETE FROM Attributes WHERE LemmaID IN (SELECT OrgLemmaID FROM Relations WHERE DateStamp >= #02/10/2004#) 2) DELETE FROM Lemmas WHERE ID IN (SELECT OrgLemmaID FROM Relations WHERE DateStamp >= #02/10/2004#) 3) DELETE FROM Relations WHERE DateStamp >= #02/10/2004#	Delete all terms entered or later linked to other terms on or after 10 February 2004.
FileID	Lemmas	1) DELETE FROM Attributes WHERE LemmaID IN (SELECT ID FROM Lemmas WHERE FileID = 1 AND PrjID = 5276148) 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT ID FROM Lemmas WHERE FileID = 1 AND PrjID = 5276148) OR DstLemmaID IN (SELECT ID FROM Lemmas WHERE FileID = 1 AND PrjID = 5276148) 3) DELETE FROM Lemmas WHERE FileID = 1 AND PrjID = 5276148	Delete all terms entered from file No. 1 of project No. 5276148.
IsSource	Lemmas	1) DELETE FROM Attributes WHERE LemmaID IN (SELECT ID FROM Lemmas WHERE IsSource = True) 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT ID FROM Lemmas WHERE IsSource = True) OR DstLemmaID IN (SELECT ID FROM Lemmas WHERE IsSource = True) 3) DELETE FROM Lemmas WHERE IsSource = True	Delete all terms entered as source terms.
Lang	Lemmas	1) DELETE FROM Attributes WHERE LemmaID IN (SELECT ID FROM Lemmas WHERE Lang = 9) 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT ID FROM Lemmas WHERE Lang = 9) OR DstLemmaID IN (SELECT ID FROM Lemmas WHERE Lang = 9) 3) DELETE FROM Lemmas WHERE Lang = 9	Delete all English (language = 9) terms.

Field	Table	SQL statements	Notes
Lemma	Lemmas	1) DELETE FROM Attributes WHERE LemmaID IN (SELECT ID FROM Lemmas WHERE Lemma LIKE '*pho*') 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT ID FROM Lemmas WHERE Lemma LIKE '*pho*') OR DstLemmaID IN (SELECT ID FROM Lemmas WHERE Lemma LIKE '*pho*') 3) DELETE FROM Lemmas WHERE Lemma LIKE '*pho*'	Delete all terms containing the letters "pho" together in that order.
PrjID	Lemmas	1) DELETE FROM Attributes WHERE LemmaID IN (SELECT ID FROM Lemmas WHERE PrjID = 5276148) 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT ID FROM Lemmas WHERE PrjID = 5276148) OR DstLemmaID IN (SELECT ID FROM Lemmas WHERE PrjID = 5276148) 3) DELETE FROM Lemmas WHERE PrjID = 5276148	Delete all terms entered from project No. 5276148.
RelTypeID	Relations	1) DELETE FROM Attributes WHERE LemmaID IN (SELECT OrgLemmaID FROM Relations WHERE RelTypeID = 2) 2) DELETE FROM Lemmas WHERE ID IN (SELECT OrgLemmaID FROM Relations WHERE RelTypeID = 2) 3) DELETE FROM Relations WHERE RelTypeID = 2	Delete all terms entered or later linked as either translations (RelTypeID = 1), synonyms (RelTypeID = 2) or antonyms (RelTypeID = 3) of other terms in an ATRIL Déjà Vu X format TDB.
SubLang	Lemmas	1) DELETE FROM Attributes WHERE LemmaID IN (SELECT ID FROM Lemmas WHERE SubLang = 2 AND Lang = 9) 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT ID FROM Lemmas WHERE SubLang = 2 AND Lang = 9) OR DstLemmaID IN (SELECT ID FROM Lemmas WHERE SubLang = 2 AND Lang = 9) 3) DELETE FROM Lemmas WHERE SubLang = 2 AND Lang = 9	Delete all British English (language = 9, sublanguage = 2) terms.
UserNick	Relations	1) DELETE FROM Attributes WHERE LemmaID IN (SELECT OrgLemmaID FROM Relations WHERE UserNick = 'Tom') 2) DELETE FROM Lemmas WHERE ID IN (SELECT OrgLemmaID FROM Relations WHERE UserNick = 'Tom') 3) DELETE FROM Relations WHERE UserNick = 'Tom'	Delete all terms entered or later linked to other terms by Tom.

Since the data relating to the term pairs is spread across three different tables (**Lemmas**, **Attributes** and **Relations**), it must be deleted from all three tables, which means using three separate **Delete statements**.

The order of the operations is extremely important: the data in the table containing the reference field must be deleted last. For example, to delete all terms entered or later linked to other terms by Tom the commands are:

- 1) DELETE FROM **Attributes** WHERE LemmaID IN (SELECT OrgLemmaID FROM **Relations** WHERE UserNick = 'Tom')
- 2) DELETE FROM **Lemmas** WHERE ID IN (SELECT OrgLemmaID FROM **Relations** WHERE UserNick = 'Tom')

3) DELETE FROM **Relations** WHERE **UserNick** = 'Tom'

Statement 1 deletes the attributes (gender, number, client, subject, etc.) of the terms entered or linked by Tom, statement 2 deletes the terms (lemmas) themselves, and statement 3 deletes the information about those terms found on the **Relations** table (which includes the translator's name). If the Delete commands are entered the other way round, the first statement would delete the translator's name so it would no longer be possible to establish which terms and attributes had been entered or linked by Tom, i.e. the other two operations would do nothing.

Since all the relations defined in ATRIL Déjà Vu X format TDBs (Translation, Synonym and Antonym) are symmetric, **OrgLemmaID** may be replaced by **DstLemmaID** in the "WHERE part" of Delete commands based on **Relations** table fields without any difference in result, unless asymmetric relations have been introduced through database customization or careless use of Update commands (see note under Update commands). If you have asymmetric relations in your database, you may have to use **DstLemmaID** instead of/combined with (using OR, AND, etc.) **OrgLemmaID**. Note that terms have **Relations** table field values for all the relations (links) they have with other terms. In other words if John entered term A as a translation of term B on 1 February 2004 and Tom "manually" linked the same term A to term C as a synonym on 15 February 2004 (on the Terminology Database Interface) the expressions for **DateStamp**, **RelTypeID** and **UserNick** shown in the table above will also delete term A even though it was originally entered at an earlier date by a different user with a different relationship! To avoid this risk, use the third **Delete statement** (DELETE FROM **Relations** WHERE...) alone, thus breaking the unwanted link, but leaving the actual terms and their attributes behind in the database.

For a full list of wildcards which may be used with the LIKE operator, see "lookup with wildcards" in the DVX on-line help or manual.

For a list of language and sublanguage codes which may be used in the **Lang** and **SubLang** fields, see "List of Language and Sublanguage Codes" in the DVX on-line help or manual.

Various Visual Basic functions may be used in Déjà Vu X SQL expressions, for example:

- 1) DELETE FROM **Attributes** WHERE **LemmaID** IN (SELECT **OrgLemmaID** FROM **Relations** WHERE **DateStamp** >= **Date()**)
- 2) DELETE FROM **Lemmas** WHERE **ID** IN (SELECT **OrgLemmaID** FROM **Relations** WHERE **DateStamp** >= **Date()**)
- 3) DELETE FROM **Relations** WHERE **DateStamp** >= **Date()**

deletes all terms entered or linked to other terms today (assuming the computer time/date setting is correct).

For a full list of Visual Basic functions see the Microsoft Access on-line help.

Example delete commands for **Attributes** table

	ATRIL Déjà Vu X TDB	Converted Déjà Vu 3 TDB	Notes
	SQL statement		
Part of Speech	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 1 AND CategoryID = 2) 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 1 AND CategoryID = 2) OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 1 AND CategoryID = 2) 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 1 AND CategoryID = 2)		Delete all verbs (see Meanings of Category IDs table on page 22)
Gender	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 2 AND CategoryID = 12) 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 2 AND CategoryID = 12) OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 2 AND CategoryID = 12) 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 2 AND CategoryID = 12)		Delete all neutral gender terms (see Meanings of Category IDs table on page 22)
Number	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 3 AND CategoryID = 13) 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 3 AND CategoryID = 13) OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 3 AND CategoryID = 13) 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 3 AND CategoryID = 13)		Delete all singular terms (see Meanings of Category IDs table on page 22)
Definition	Field not present	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 4 AND TextValue LIKE '*pho*') 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 4 AND TextValue LIKE '*pho*') OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 4 AND TextValue LIKE '*pho*') 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 4 AND TextValue LIKE '*pho*')	Delete all terms whose definitions contain the letters "pho" together in that order

	ATRIL Déjà Vu X TDB	Converted Déjà Vu 3 TDB	
	SQL statement		Notes
Subject	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 4 AND TextValue = '75') 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 4 AND TextValue = '75') OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 4 AND TextValue = '75') 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 4 AND TextValue = '75')	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 5 AND TextValue = '75') 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 5 AND TextValue = '75') OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 5 AND TextValue = '75') 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 5 AND TextValue = '75')	Delete terms associated with Subject No. 75
Client	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 5 AND TextValue = '23') 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 5 AND TextValue = '23') OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 5 AND TextValue = '23') 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 5 AND TextValue = '23')	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 6 AND TextValue = '23') 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 6 AND TextValue = '23') OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 6 AND TextValue = '23') 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 6 AND TextValue = '23')	Delete terms associated with Client No. 23
Context	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 6 AND TextValue LIKE '*pho*') 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 6 AND TextValue LIKE '*pho*') OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 6 AND TextValue LIKE '*pho*') 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 6 AND TextValue LIKE '*pho*')	1) DELETE FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 7 AND TextValue LIKE '*pho*') 2) DELETE FROM Relations WHERE OrgLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 7 AND TextValue LIKE '*pho*') OR DstLemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 7 AND TextValue LIKE '*pho*') 3) DELETE FROM Attributes WHERE LemmaID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 7 AND TextValue LIKE '*pho*')	Delete all terms whose context contains the letters "pho" together in that order

Also see notes before and after **Example delete commands for Lemmas and Relations tables** chart.

Update commands

To enter Update commands in the Terminology Database interface, choose "Execute SQL..." from the "Database" menu. To see the effects of a command you must refresh the display by closing and reopening the database.

It is advisable to repair and compact the database after updating data to improve efficiency and make sure the changes take effect ("Tools" menu).

Example update commands for Lemmas and Relations tables

Field	Table	SQL statement	Notes
DateStamp	Relations	UPDATE Relations SET DateStamp = #02/10/2004# WHERE Select statement	Change the date on which the terms identified by the Select statement were entered or linked to 10 February 2004.
FileID	Lemmas	UPDATE Lemmas SET FileID = 2 WHERE Select statement	Change the file from which the terms identified by the Select statement were entered in their respective projects to No. 2.
IsSource	Lemmas	UPDATE Lemmas SET IsSource = True WHERE Select statement	Mark the terms identified by the Select statement as having originally been entered as source terms.
Lang	Lemmas	UPDATE Lemmas SET Lang = 9 WHERE Select statement	Change the language associated with the terms identified by the Select statement to English (language No. 9).
Lemma	Lemmas	UPDATE Lemmas SET Lemma = 'hello' WHERE Select statement	Change the terms identified by the Select statement to "hello".
PrjID	Lemmas	UPDATE Lemmas SET PrjID = 5276148 WHERE Select statement	Change the project from which the terms identified by the Select statement were entered into No. 5276148.
RelTypeID	Relations	UPDATE Relations SET RelTypeID = 1 WHERE Select statement	Change all terms identified by the Select statement into translations (RelTypeID = 1), synonyms (RelTypeID = 2) or antonyms (RelTypeID = 3) in an ATRIL Déjà Vu X format TDB.
SubLang	Lemmas	UPDATE Lemmas SET SubLang=2 WHERE Select statement	Change the language associated with the terms identified by the Select statement to sublanguage No. 2 of their respective languages.
UserNick	Relations	UPDATE Relations SET UserNick = 'Tom' WHERE Select statement	Mark the terms identified by the Select statement as having been entered or linked by Tom.

The Select statement may be any of the statements (or combination thereof) shown on the **Example select statements** chart, but the short form can only be used if the field in the Select statement is on the same table as the field to be updated, for example:

```
UPDATE Lemmas SET SubLang=2 WHERE Lang = 9
(SubLang and Lang are both on the Lemmas table, therefore the short form may be used)
```

```
UPDATE Lemmas SET SubLang=2 WHERE ID IN (SELECT OrgLemmaID FROM Relations WHERE UserNick ='Tom')
(SubLang is on the Lemmas table, but UserNick is on the Relations table, therefore the long form is needed)
```

It is advisable to make sure the [Select statement](#) gives the expected result by trying it alone in the Terminology Database interface first (where possible). Bear in mind however that [Select statements](#) in the Terminology Database interface are ANDed with the language chosen as Source Language (see note under **SQL Select statements**).

If no [Select statement](#) is added, i.e.:

```
UPDATE Lemmas SET SubLang=2
```

all the terms in the Terminology Database will be set to sublanguage No. 2.

The field in the [Select statement](#) can also be the same as the field which needs updating, for example:

```
UPDATE Lemmas SET SubLang=2 WHERE SubLang=1
```

corrects the sublanguage from 1 to 2.

You can update several fields **on the same table** at once in this way:

```
UPDATE Lemmas SET SubLang=2, Lang=9 WHERE PrjID = 5276148
```

When using Update commands on [Relations](#) table fields, be careful not to introduce spurious asymmetric links (relations) between terms. Example: when John entered the term "telephone" (English) with its corresponding translation "telefono" (Italian) on 10 February 2004, DVX set up two symmetric links (relations): "telephone is the translation of telefono" and "telefono is the translation of telephone". If Tom enters this Update command:

```
UPDATE Relations SET UserNick = 'Tom', DateStamp = #03/10/2004# WHERE ID IN (SELECT ID FROM Lemmas WHERE Lemma LIKE '*pho*')
```

the relation "telephone is the translation of telefono" will seem to have been input by Tom on 10 March 2004, whereas "telefono is the translation of telephone" will continue to be associated with John and 10 February 2004. This will probably not cause any serious

problems until someone else enters a Delete or another Update command involving **Relations** table fields, which will then have almost unpredictable results. To avoid this risk, base your Update command on a common property of both the source and target terms.

For a list of language and sublanguage codes which may be used in the **Lang** and **SubLang** fields, see "List of Language and Sublanguage Codes" in the DVX on-line help or manual.

Various Visual Basic functions may be used in Déjà Vu X SQL commands, for example:

```
UPDATE Relations SET DateStamp = Date()
```

marks all terms as having been entered or linked today (assuming the computer time/date setting is correct).

For a full list of Visual Basic functions see the Microsoft Access on-line help.

Example update commands for **Attributes** table

	ATRIL Déjà Vu X TDB	Converted Déjà Vu 3 TDB	Notes
	SQL statement		
Part of Speech	UPDATE Attributes SET CategoryID = 2 WHERE AttrTypeID =1 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)		Change all the terms identified by the Select statement to verbs (see Meanings of Category IDs table on page 22)
Gender	UPDATE Attributes SET CategoryID = 12 WHERE AttrTypeID =2 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)		Change all the terms identified by the Select statement to neutral gender (see Meanings of Category IDs table on page 22)
Number	UPDATE Attributes SET CategoryID = 13 WHERE AttrTypeID =3 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)		Change all the terms identified by the Select statement to singular (see Meanings of Category IDs table on page 22)
Definition	Field not present	UPDATE Attributes SET TextValue ='meaningless word' WHERE AttrTypeID =4 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)	Change the definition associated with the terms identified by the Select statement to "meaningless word"
Subject	UPDATE Attributes SET TextValue ='75' WHERE AttrTypeID =4 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)	UPDATE Attributes SET TextValue ='75' WHERE AttrTypeID =5 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)	Change the subject associated with the terms identified by the Select statement to No. 75

	ATRIL Déjà Vu X TDB	Converted Déjà Vu 3 TDB	
	SQL statement		Notes
Client	UPDATE Attributes SET TextValue ='24' WHERE AttrTypeID =5 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)	UPDATE Attributes SET TextValue ='24' WHERE AttrTypeID =6 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)	Change the client associated with the terms identified by the Select statement to No. 24
Context	UPDATE Attributes SET TextValue ='very boring text' WHERE AttrTypeID =6 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)	UPDATE Attributes SET TextValue ='very boring text' WHERE AttrTypeID =7 AND LemmaID IN (SELECT ID FROM Lemmas WHERE Select statement)	Change the context associated with the terms identified by the Select statement to "very boring text"

You cannot update an attribute that does not already exist. If you want to add a missing attribute use an **Insert command** (see next section). If the attribute you wish to update exists in some cases, but is missing in others, delete the attribute concerned where it exists (use a **Delete command**) then use an **Insert command**.

The **Select statement** for **Attributes** table fields deserves special attention, for example:

```
UPDATE Attributes SET TextValue='24' WHERE AttrTypeID=5 AND LemmaID IN (SELECT ID FROM Lemmas WHERE ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID = 4 and TextValue = '75'))
```

changes the value of attribute No. 5 to "24" (Client No. 24 in an ATRIL Déjà Vu X format TDB) where attribute No. 4 is equal to 75 (Subject No. 75 in an ATRIL Déjà Vu X format TDB)

If the **Select statement** field is the same as the field which needs updating, the expression becomes much simpler:

```
UPDATE Attributes SET TextValue='24' WHERE AttrTypeID=5 AND TextValue='23'
```

corrects the value of attribute No. 5 from "23" to "24" (Client in an ATRIL Déjà Vu X format TDB).

Also see notes before and after **Example update commands for Lemmas and Relations tables** chart.

Insert commands

To enter Insert commands in the Terminology Database interface, choose "Execute SQL..." from the "Database" menu. To see the effects of a command you must refresh the display by closing and reopening the database. It is advisable to repair and compact the database after inserting data to improve efficiency and make sure the changes take effect ("Tools" menu).

Example insert commands for **Attributes** table

	ATRIL Déjà Vu X TDB	Converted Déjà Vu 3 TDB	Notes
	SQL statement		
Part of Speech	INSERT INTO Attributes SELECT 2 AS CategoryID , 1 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement		Mark all the terms identified by the Select statement as verbs
Gender	INSERT INTO Attributes SELECT 12 AS CategoryID , 2 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement		Mark all the terms identified by the Select statement as neutral gender
Number	INSERT INTO Attributes SELECT 13 AS CategoryID , 3 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement		Mark all the terms identified by the Select statement as singular
Definition	Field not present	INSERT INTO Attributes SELECT 'meaningless word' AS TextValue , 4 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement	Add the definition "meaningless word" to all the terms identified by the Select statement
Subject	INSERT INTO Attributes SELECT '75' AS TextValue , 4 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement	INSERT INTO Attributes SELECT '75' AS TextValue , 5 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement	Mark the terms identified by the Select statement as being associated with subject No. 75
Client	INSERT INTO Attributes SELECT '24' AS TextValue , 5 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement	INSERT INTO Attributes SELECT '24' AS TextValue , 6 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement	Mark the terms identified by the Select statement as being associated with client No. 75
Context	INSERT INTO Attributes SELECT 'very boring text' AS TextValue , 6 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement	INSERT INTO Attributes SELECT 'very boring text' AS TextValue , 7 AS AttrTypeID , ID AS LemmaID FROM Lemmas WHERE Select statement	Add the context "very boring text" to all the terms identified by the Select statement

Insert commands may be used on the **Attributes** table to insert missing attributes, for example after importing an external glossary which did not include a particular attribute. This following **Select statement** will find terms without attribute No. 5 (Client in an ATRIL Déjà Vu X format TDB):

```
NOT ID IN (SELECT LemmaID FROM Attributes WHERE AttrTypeID=5)
```

Warning: do not try to insert an attribute which already exists. Use an **Update command** to change it.

Also see notes before and after **Example update commands for Lemmas and Relations tables** chart.