VAMDC Query Language Specification

Document Information

Editors:	L. Nenadovic, G. Rixon, M. Doronin
Authors:	G. Rixon
Contributors:	Thomas Marquart

Type of document:	standards documentation
Status:	draft
Distribution:	public
Work package:	WP6
Version:	11.12
Date:	21/12/2011
Document code:	
Document URL:	http://www.vamdc.org/documents/querylanguage_v11.12.pdf

Abstract: This document describes the VAMDC SQL subset query language, both versions 1 and 2. Both versions should be supported by VAMDC nodes. Version 2 is a superset of version 1, so any valid VSS1 query is a valid VSS2 query.

Version History

Version	Date	Modified By	Description of Change
V0.1	11/01/2011	G. Rixon	first draft
V11.5	27/05/2011	L. Nenadovic	editing for first release
V11.12	21/12/2011	G. Rixon	Definition of the VSS2 specification

Acknowledgements

VAMDC is funded under the "Combination of Collaborative Projects and Coordination and Support Actions" Funding Scheme of The Seventh Framework Program. Call topic: INFRA-2008-1.2.2 Scientific Data Infrastructure. Grant Agreement number: 239108.

CONTENTS

CHAPTER

ONE

VSS1

1.1 Introduction

VAMDC SQL Subset 1 (VSS1) is a query language designed for the VAMDC-TAP web-services. A query in VSS1 defines an extract of an archive that a data service can return in an VAMDC-XSAMS document or in a tabular format.

1.1.1 Subset of SQL92

VSS1 is based syntactically on ISO SQL92 but discards almost all the features of that language. The only features remaining are those relevant to selecting data to build an VAMDC-XSAMS document; VSS1 contains no feature for modifying a database. VSS1 assumes VAMDC's standard view of the database as a single table. This latter point means that there is only one visible table (which therefore does not need to be named in a query) and the column names are terms taken from the VAMDC dictionary. This is an example of a VSS1 query:

```
SELECT ALL WHERE AtomNuclearCharge = 25 AND AtomIonCharge < 2
```

In the following definitions, VSS1 queries are assumed to be submitted by a client application to a web-service. That service contains a query processor that converts the VSS1 query into a SQL query or set of queries, suitable for the relational database managed by the service. A request to the service contains the VSS1 text of the query and other parameters such as the desired formats of the returned results.

1.1.2 Interoperability and extensions

VSS1 is intended to be interoperable across all VAMDC databases. The same query can be accepted at all database, even though different data are raised from each.

Making VSS1 fully interoperable and capable of easy implementation would exclude all the higher functions of SQL (mainly features of the WHERE clause, such as sub-queries). Where these features can easily be added to a particular query-processor it would be a beneficial to have them. Therefore, certain parts of SQL92 are noted as extensions to VSS1. A query containing extensions is not strictly valid VSS1 and therefore not interoperable across all database; it may work on particular database. Users and applications should normally use only valid VSS1, but may use extensions on queries written specially for individual databases.

1.2 Narrative definition (normative)

To be a valid in VSS1, a query must satisfy the syntactic rules of the SQL92 language.

VSS1 queries never alter the database to which they are applied.

- VSS1 query must be a SELECT statement.
- VSS1 queries must not contain the SQL92 keywords ALTER, CREATE, DELETE, DROP, INSERT, REPLACE or UPDATE.

- VSS1 queries must not contain the SELECT ... INTO construction of SQL92.

VSS1 queries must not contain the JOIN keyword.

Column names used as operands in a VSS1 query must be terms taken from the VAMDC Restrictables dictionary. These columns names must not be qualified by the name of a schema or database.

All the terms in the dictionary are valid as column names on all databases with a VSS1 processor. The query processor must implement the translation of the dictionary terms to names of real columns in the underlying database.

VSS1 processors may accept only a sub-set of the dictionary keywords, corresponding to the content of the underlying database. This sub-set naturally varies between databases and the set of restrictables for a given database is normally made available to the clients of the database. Where a query includes restrictables not supported by a given VSS1 processor, the processor must reject the query; it must not process the query while ignoring the unsupported constraints.

When the results of a query are to be returned in VAMDC-XSAMS format, VSS1 queries should begin with SELECT ALL ...; the set of columns from which data are returned is implicitly chosen by the choice of VAMDC-XSAMS format. If such a query does specify a set of columns (e.g. SELECT AtomIonCharge WHERE ...), then the query processor should ignore that set and proceed as if the query were SELECT ALL. However, where the results of a query are to be returned in a tabular format, the query processor must respect the query's selection of columns. In the latter case, if the query specifies a returnable not supported by the particular database then the processor should reject the query.

When processing a query that contains valid VSS1 plus extensions, the behaviour is defined by the implementation of the query processor. The processor may reject the query, or it may ignore the extensions that it does not support.

The following parts of SQL92 constitute VSS1 extensions: EXISTS, GROUP BY, HAVING, UNION, INTERSECT, EXCEPT, MINUS, ORDER BY, LIMIT, DECLARE, FETCH, CLOSE.

1.3 SQL92 syntax (informative)

The SQL92 standard [SQL92] should be consulted for the normative rules of syntax. These notes are for easy reference. VSS1 excludes so much of SQL that only the low-level aspects of the syntax are relevant.

SQL queries are written as text strings containing keywords, operators and operands separated by white space. Operands are names of tables and columns, sometimes called SQL identifiers or literal values. Identifiers and literals are sensitive to case; keywords and operators are not. There is a convention of writing keywords in upper case.

Queries can contain any Unicode character, but the keywords can be written using only ASCII characters. In VSS1, the valid identifiers also use only ASCII characters.

White space is required between keywords and operands but not between operators and operands. A typical (simple) VSS1 query looks like this:

```
SELECT ALL WHERE AtomIonCharge>6
```

This query would be equally valid:

```
SELECT * WHERE AtomIonCharge > 6
```

Here, data are selected from the columns AtomIonCharge and AtomNuclearCharge (note the use of a comma-separated list to specify the columns) of the table States according to a criterion on the electronic charge of the ions. String literals are delimited by single quotes (the ASCII apostrophe character) thus:

```
... WHERE AtomSymbol='Fe' ...
```

To include an apostrophe in a string, write two consecutive apostrophe-characters. If an identifier contains 'special characters' (typically white space), it must be protected with double quotes thus:

```
SELECT * WHERE "silly column name" > 0...
```

CHAPTER

TWO

VSS2

2.1 Introduction

VAMDC SQL Subset 2 (VSS2) is a query language designed for the VAMDC-TAP web-services. A query in VSS2 defines an extract of an archive that a data service can return in an VAMDC-XSAMS document or in a tabular format.

2.1.1 Subset of SQL92

VSS2 is based syntactically on ISO SQL92 but discards almost all the features of that language. The only features remaining are those relevant to selecting data to build an VAMDC-XSAMS document; VSS2 contains no feature for modifying a database. VSS2 assumes VAMDC's standard view of the database as a single table. This latter point means that there is only one visible table (which therefore does not need to be named in a query) and the column names are terms taken from the VAMDC dictionary. This is an example of a VSS2 query:

```
SELECT Species WHERE AtomNuclearCharge = 25 AND AtomIonCharge < 2
```

In the following definitions, VSS2 queries are assumed to be submitted by a client application to a web-service. That service contains a query processor that converts the VSS2 query into a SQL query or set of queries, suitable for the relational database managed by the service. A request to the service contains the VSS2 text of the query and other parameters such as the desired formats of the returned results.

2.1.2 Superset of VSS1

VSS2 is a superset of VAMDC SQL Subset 1 (VSS1). It adds several aspects to VSS1:

- ability to specify the set of XSAMS branches to be returned, like Molecules, States, etc.
- ability to group restrictables related to different transition's states and different products/reactants of the same reaction.

XSAMS branches selection

Keywords, defining the desired branches, may be specified in SELECT part of the query, like:

```
SELECT molecules where MoleculeStoichiometricFormula = "C10H20"
```

this query should return only a list of molecules, including references and excluding all the state-related or process-related information.

If query specifies a states or quantum numbers branch, parent species information should also be provided. Contrary the selection of species should not cause the output of states or quantum numbers.

For the full list of possible branch selectors, their behaviour and relations see the **Requestables** section of the VAMDC dictionary

Restrictables prefixes

Another addition is the prefixes for restrictables keywords, used for processes selection refinement.

- When selecting transition information, prefices to the state-related restrictables are grouping them by upper or lower state. Examples:

  ```
  Select * where lower.StateEnergy = 0 and upper.StateEnergy > 1000
  select * where StateEnergy = 0 and upper.StateEnergy > 1000
  ```

  Note: query

  ```
  select * where StateEnergy < 100 and lower.StateEnergy>100
  ```

  will (and must) return no results.

  If the prefix is omitted in the query, constraints apply to both states of a transition.

- When selecting collision information, prefices are **reactantX** and **productX**, where X is a case-insensitive alphanumeric symbol [0-9A-Z] , defining a group of keywords applying to the single reactant. For the special case of two-particle collisions, in which it is necessary to distinguish the incident particle, the prefices **target** and **collider** may be used instead.

  Example:

  ```
  Select collisions,states where reactantA.AtomSymbol = "O" and reactantA.AtomIonCharge = 1 and
  ```

  would return all reactions data between O^+ atomic ion and *NH*, *CH* or *C2* molecules

  Another example:

  ```
  Select collisions,states where reactantA.AtomSymbol = "O" and ( reactantA.AtomIonCharge = 1 O
  ```

  should return all reactions data of one or two times ionized oxygen atom.

2.1.3 Interoperability and extensions

VSS2 is intended to be interoperable across all VAMDC databases. The same query can be accepted at all database, even though different data are raised from each.

Making VSS2 fully interoperable and capable of easy implementation would exclude all the higher functions of SQL (mainly features of the WHERE clause, such as sub-queries). Where these features can easily be added to a particular query-processor it would be a beneficial to have them. Therefore, certain parts of SQL92 are noted as extensions to VSS2. A query containing extensions is not strictly valid VSS2 and therefore not interoperable across all database; it may work on particular database. Users and applications should normally use only valid VSS2, but may use extensions on queries written specially for individual databases.

2.2 Narrative definition (normative)

To be a valid in VSS2, a query must satisfy the syntactic rules of the SQL92 language.

VSS2 queries never alter the database to which they are applied.

- VSS2 query must be a SELECT statement.
- VSS2 queries must not contain the SQL92 keywords ALTER, CREATE, DELETE, DROP, INSERT, REPLACE or UPDATE.
- VSS2 queries must not contain the SELECT ... INTO construction of SQL92.

VSS2 queries must not contain the JOIN keyword.

Column names used as operands in a VSS2 query must be terms taken from the VAMDC dictionary. Column names may be written in any mix of upper and lower case and query processors must treat all variations of case as equivalent.

Column names appearing in the WHERE clause must be taken from the Restrictables dictionary. These names may be qualified by a context, e.g. to distinguish between the upper and lower states of an electronic transition. The qualified name is written with the context name as a prefix to the restrictable name, separated by a full stop, e.g. `upper.StateEnergy`. The following contexts are defined in VSS2.

- `reactantx` (where x is any single character) associates column names with the nth reactant in a chemical network. E.g. `reactant1.MoleculeInchiKey`.
- `productx` (where x is any single character) associates column names with the nth product in a chemical network. E.g. `product2.MoleculeInchiKey`.
- `collider` associates column names with the incident particle in a collision. E.g. `collider.AtomSymbol`.
- `target` associates column names with the target particle in a collision. E.g. `target.StateEnergy`.
- `upper` associates column names with the higher-energy state of a transition. E.g. `upper.StateEnergy`.
- `lower` associates column names with the lower-energy state of a transition. E.g. `lower.StateLifeTime`.

Context prefices may be written in any mix of upper and lower case and query processors must treat all variations of case as equivalent. This includes the final character in the `reactantx` and `productx` prefices: `reactantA` must be treated as equivalent to `reactanta`.

The list of column names following the SELECT keyword, which specify the columns from which data are to be returned, must be taken from the Requestables dictionary, or must contain only the single keyword ALL (that keyword having its normal meaning in SQL92). Note that the 'columns' in this dictionary are composites. In a tabular representation of the results a requestable 'column' may produce multiple output-columns. In an XSAMS representation, a requestable 'column' may produce an XML fragment with significant sub-structure. Column names may be written in any mix of upper and lower case and query processors must treat all variations of case as equivalent.

The query processor must implement the translation of the dictionary terms to names of real columns in the underlying database.

VSS2 processors may accept only a sub-set of the dictionary keywords, corresponding to the content of the underlying database. This sub-set naturally varies between databases and the set of restrictables and requestables for a given database is normally made available to the clients of the database. Where a query includes restrictables or requestables not supported by a given VSS2 processor, the processor must reject the query; it must not process the query while ignoring the unsupported items.

A VSS2 processor should accept only the (possibly empty) sub-set of the context prefices that apply to its database. E.g. processors that have no data on reactions should reject the `reactantx` prefix.

When processing a query that contains valid VSS2 plus extensions, the behaviour is defined by the implementation of the query processor. The processor may reject the query, or it may ignore the extensions that it does not support.

The following parts of SQL92 constitute VSS2 extensions: EXISTS, GROUP BY, HAVING, UNION, INTERSECT, EXCEPT, MINUS, ORDER BY, LIMIT, DECLARE, FETCH, CLOSE.

2.3 SQL92 syntax (informative)

The SQL92 standard [SQL92] should be consulted for the normative rules of syntax. These notes are for easy reference. VSS2 excludes so much of SQL that only the low-level aspects of the syntax are relevant.

SQL queries are written as text strings containing keywords, operators and operands separated by white space. Operands are names of tables and columns, sometimes called SQL identifiers or literal values. Identifiers and

literals are sensitive to case; keywords and operators are not. There is a convention of writing keywords in upper case.

Queries can contain any Unicode character, but the keywords can be written using only ASCII characters. In VSS2, the valid identifiers also use only ASCII characters.

White space is required between keywords and operands but not between operators and operands. A typical (simple) VSS2 query looks like this:

```
SELECT ALL WHERE AtomIonCharge>6
```

This query would be equally valid:

```
SELECT ALL WHERE AtomIonCharge > 6
```

Here, data are selected from the columns AtomIonCharge and AtomNuclearCharge (note the use of a comma-separated list to specify the columns) of the table States according to a criterion on the electronic charge of the ions. String literals are delimited by single quotes (the ASCII apostrophe character) thus:

```
... WHERE AtomSymbol='Fe' ...
```

To include an apostrophe in a string, write two consecutive apostrophe-characters. If an identifier contains 'special characters' (typically white space), it must be protected with double quotes thus:

```
SELECT "silly column name" WHERE...
```

CHAPTER

THREE

REFERENCES

[SQL92] Information Technology - Database Language SQL (Proposed revised text of DIS 9075) July 1992 http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt

CHAPTER

FOUR

CHANGE LOG FOR QUERY LANGUAGE

4.1 Changes leading up to v11.05

The query-language "VAMDC SQL Sub-set 1" (VSS1) was defined. This language was the only one formally supported by VAMDC data-services in the 11.05 release of the system.

4.2 Changes in the 11.09 system-release

No new version of the standard was issued for this system release, but informal support for a new query language, VSS2, was implemented in some nodes.

VSS2 is VAMDC SQL Sub-set 2 and is a super-set of VSS1. Nodes that can process VSS2 queries can process VSS1 queries using the same code.

The features in VSS2 but not in VSS1 are:

- support for "requestables" to restrict the parts of the data model returned in the query results;
- prefices on restrictables to group them by context; the prefices at this time were "initial" and "final", relating to states involved in transitions, "reactantn" and "productn" relating to collisions.

4.3 Changes between v11.09 and v11.12

VSS2 was formally defined. Its specification document is independent of that for VSS1, even though the two languages are related.

The prefices on restrictables were changed. "Upper" and "lower" replaced "initial" and "final".

In v11.12 of the system, VAMDC nodes are expected to accept either VSS1 or VSS2.