

How-To Guide  
Document Version: 2016-10-13

PARTNER

# How to Convert SQL from the Microsoft SQL Server Database to the SAP HANA™ Database

All Countries

# Typographic Conventions

Type Style	Description
<i>Example</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
<b>Example</b>	Emphasized words or expressions.
<b>EXAMPLE</b>	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
<b>Example</b>	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
< <b>Example</b> >	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
<b>EXAMPLE</b>	Keys on the keyboard, for example, <b>F2</b> or <b>ENTER</b> .

# Document History

Date	Change
2014-07-24	<p>The converter was changed in the following ways:</p> <ul style="list-style-type: none"><li>• If SAP HANA does not support certain SQL, and no equivalent is available in the SAP HANA database, the SQL will be replaced by <code>### UNSUPPORTED STATEMENT ###</code> or <code>### UNSUPPORTED ###</code> in the output file.</li><li>• You can now convert SQL with non-standard syntax (tokens). For more information, see <a href="#">Converting SQL with Non-Standard Syntax (Tokens)</a>.</li><li>• Wider scope of supported "date" formats and string concatenation.</li></ul>
2016-09-29	<p>The converter was changed in the following ways:</p> <ul style="list-style-type: none"><li>• The converter supports the WITH statement.</li><li>• There is one more default pattern in the token file. For more information, see <a href="#">Token File</a>.</li><li>• Both 32-bit and 64-bit versions are available for the converter.</li><li>• There is one more API example for the 64-bit version. For more information, see <a href="#">API Example</a>.</li></ul>

---

# Table of Contents

1	Introduction.....	5
2	Using the SQL Converter .....	6
2.1	Configuration File .....	8
2.2	Command-Line Parameters .....	9
2.3	Converting SQL with Non-Standard Syntax (Tokens) .....	11
2.3.1	Token File.....	12
3	Using the SQL Converter via API.....	13
3.1	API Example .....	14
4	Identifiers .....	15
5	Supported Data Types .....	16
6	Supported Operators.....	18
7	Supported Expressions.....	19
8	Supported SQL Statements .....	20
8.1	Supported DML.....	20
8.2	Supported DDL .....	20
8.3	Supported Complex Statements .....	21
9	Supported Features with Limitation.....	22
9.1	Concatenating Strings .....	22
9.2	Converting Date Formats.....	22
9.2.1	Using String as DATE Parameter.....	22
9.2.2	Using Integer Value as DATE Parameter .....	23

# 1 Introduction

This guide describes how to convert structured query language (SQL) in the Microsoft SQL Server database (using T-SQL grammar) to SQL that can be used in the SAP HANA™ database (using ANSI-SQL grammar) using the SQL converter.

The SQL converter is a semi-automatic tool that helps convert most of the data-definition language (DDL) and data-manipulation language (DML). After the conversion, you must check whether the converted version is correct according to your needs.

This tool supports most of the official T-SQL grammar, as well as some well-known and widely-used undocumented features. For more information about the official T-SQL grammar, see the MSDN Library at <http://msdn.microsoft.com/en-us/library>.

If SAP HANA does not support certain SQL, this tool will find equivalents in the SAP HANA database and convert the SQL. If no equivalent is available, the SQL will be replaced by `### UNSUPPORTED STATEMENT ###` or `### UNSUPPORTED ###` in the output file.



## Example

The input file is as follows:

```
Select DateDiff("yy", '20100101', '20141201'), DateDiff(qq, '20100101', '20140601');
```

After conversion, the output file is as follows:

```
SELECT YEAR('2014-12-01') - YEAR('2010-01-01'), ### UNSUPPORTED ### FROM DUMMY;
```



## Note

If you encounter any issue when using the SQL converter, post the issue to the SAP Community Network at <http://scn.sap.com/community/business-one> for general topics or at <http://scn.sap.com/community/business-one/system-administration> for system-administration topics.

## 2 Using the SQL Converter

The tool converts the input file (SQL in the Microsoft SQL Server database) to an output file (SQL that you can use in the SAP HANA database).

You can run the SQL converter using either the command line or the GUI mode.

### Note

The SQL converter converts only the SQL that you have run successfully in the Microsoft SQL Server database.

### Prerequisites

- You have installed .Net Framework 4.0 or later on the computer on which you want to use the SQL converter.
- You have installed the SAP HANA database client software on the computer on which you want to use the SQL converter. For more information, see *SAP HANA Client Installation and Update Guide* on SAP Help Portal at [http://help.sap.com/hana\\_appliance](http://help.sap.com/hana_appliance).

### Note

The client must be the same version as the SQL converter you want to use. For example, for the 32-bit SQL converter, you must install the 32-bit SAP HANA database client, and for the 64-bit SQL converter, you must install the 64-bit SAP HANA database client.

- You have downloaded the SQL converter *.zip* file.
- You have ensured that the SQL you want to convert can run successfully in the Microsoft SQL Server database.

### Procedure

- To run the SQL converter using the command line, follow the procedure below:
  1. Extract the SQL converter *.zip* file to anywhere on your computer.
  2. In the Microsoft Windows command window, navigate to the SQL converter folder. For more information about Microsoft Windows commands, see the command-line reference in the Microsoft Windows on-line help.
  3. Enter `Converter.exe <command-line parameters>`.

If you do not use a certain command-line parameter, the system runs the tool according to the settings you defined for that parameter in the configuration file. Parameters in the command line overwrite the parameters you defined in the configuration file. For more information, see [Configuration File](#) and [Command-Line Parameters](#).

## Note

If there are multiple statements in the input file, separate them with a semicolon ";". After the conversion, the converted statements are listed in the output file according to the original sequence.

For example, the input file is as follows:

```
select CardCode, CardName from OCRD where CardType = 'C';
select max(DocEntry) from ORDR;
```

After conversion, the output file is as follows:

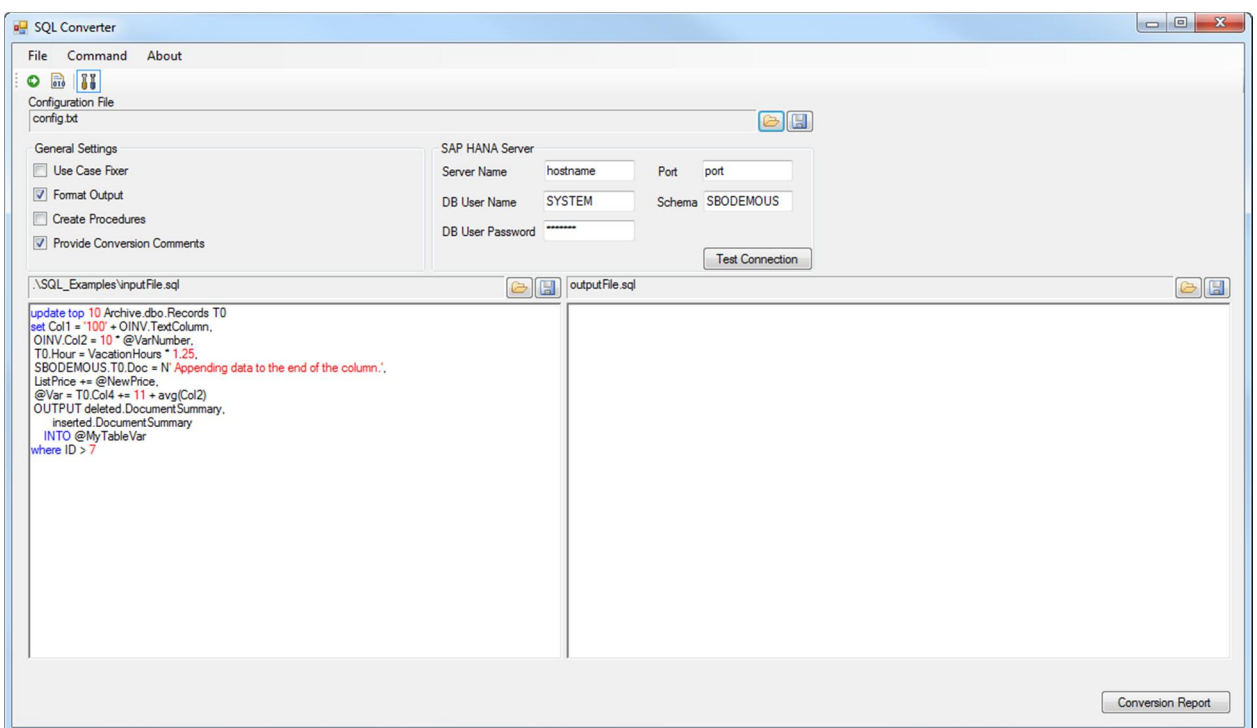
```
SELECT "CardCode", "CardName" FROM OCRD WHERE "CardType" = 'C';
SELECT MAX("DocEntry") FROM ORDR;
```

- To run the SQL converter using the GUI mode, follow the procedure below:

1. Extract the SQL converter .zip file to anywhere on your computer.
2. In the SQL converter folder, double-click *Converter.exe*.

Alternatively, in the Microsoft Windows command window, navigate to the SQL converter folder, and enter **Converter.exe -g**.

The *SQL Converter* window appears.



For more information about the settings, see [Command-Line Parameters](#).



3. In the *SQL Converter* window, select a configuration file. By default, the *Config.txt* file in the SQL converter folder is selected. For more information about the configuration file, see [Configuration File](#) and [Command-Line Parameters](#).

You can also modify the settings in the *General Settings* area when you are running the converter.

4. In the *SQL Converter* window, select an input file, or enter the SQL that you want to convert.


---

5. To run the converter, in the *SQL Converter* window, perform one of the following:

- Choose the  (*Run Converter*) icon.
- Choose *Command* → *Run Converter*.
- Choose the  keyboard button.

The SQL in the input file or in the input box is converted, and the *Conversion Report* window appears.

You can find the converted SQL in the output box.

6. To save the converted SQL to an output file, choose the  (*Save*) icon beside the *Output File* field.

## Result

The defined SQL is converted from T-SQL grammar to ANSI-SQL grammar. You must perform checks on the converted version to ensure that it is correct according to your needs.

## 2.1 Configuration File

You can use the *Config.txt* file to store parameters that you do not change frequently. However, the parameters in the command line overwrite the parameters you defined in the configuration file. If you do not use a certain command-line parameter, the system runs the tool according to the settings you defined for that parameter in the configuration file.


In the configuration file, explanations for parameters are preceded by *//*.

For more information about the parameters, see [Command-Line Parameters](#).

### Note

If the configuration file contains multiple lines or values for the same parameter, the last line or value is used.



 Example

An example of the configuration file is as follows:

```
// If UseCaseFixer=true, DB objects (tables, columns,...) will be verified.
// If UseCaseFixer=false, DB objects will NOT be verified but the SQL converter will run faster.
UseCaseFixer=false
// Case fixer will use the following DB settings:
DBServer=10.55.178.115:30915
DBSchema=SBODEMOUS
DBUser=SYSTEM
DBPasswd=Manager111
// End of DB settings for the case fixer
// Exclude conversion comments from the output file or not
DisableComments=false
// Specify names for the input and output files
InputFile= c:\WorkDir\Ppresentations\HANATranslator_0912\Examples.sql
OutputFile= c:\WorkDir\Ppresentations\HANATranslator_0912\output.sql
// Format the output file in clean and professional SQL layout or not
Formatter=false
// Definition of tokens
TokenFile= c:\WorkDir\Ppresentations\HANATranslator_0912\tokenFile.txt
// Start using GUI
GUI=true
```

## 2.2 Command-Line Parameters

You can use the following parameters in the command line to overwrite those defined in the configuration file. If you do not use a certain command-line parameter, the system runs the tool according to the settings you defined for that parameter in the configuration file.

### Command-Line Parameters

Parameter	Equivalent in Configuration File	Description	Example
f	UserCaseFixer	Enables the case fixer to check whether all tables mentioned in the input file exist in the SAP HANA database. The default value is false.	-f

Parameter	Equivalent in Configuration File	Description	Example
s	DBServer	Address and port number of the SAP HANA database server	<i>-s 10.55.178.115:30915</i>
d	DBSchema	Schema name in the SAP HANA database	<i>-d SBODEMOUS</i>
u	DBUser	User name to log on to the SAP HANA database	<i>-u SYSTEM</i>
p	DBPasswd	User password to log on to the SAP HANA database	<i>-p Manager111</i>
c	DisableComments	Excludes conversion comments from the output file. The default value is false. Conversion comments in the output file explain which statements were converted, and which were not, providing reasons.	<i>-c</i>
i	InputFile	Path of the input file	<i>-i c:\HANATranslator_0912\input.sql</i>
o	OutputFile	Path of the output file	<i>-o c:\HANATranslator_0912\output.sql</i>
F	Formatter	Formats the output file in a clean and professional SQL layout. The default value is false.	<i>-F</i>
t	TokenFile	Path of the token file	<i>-t c:\HANATranslator_0912\tokenFile.txt</i>
g	GUI	Enables the GUI mode of the SQL converter. The default value is true.	<i>-g</i>
h	Help (does not exist in the file)	Displays help for command-line parameters	<i>-h</i>
P	Create procedures (does not exist in the file)	Enables automatic creation of stored procedures in cases of queries with context dependencies. The default value is false.	<i>-P</i>
v	Version (does not exist in the file)	Displays the name and version of the tool	<i>-v</i>

## 2.3 Converting SQL with Non-Standard Syntax (Tokens)

To increase flexibility, SQL is often saved with tokens. Those tokens are replaced by other SQL content when you run the saved SQL.

### Prerequisites

- You have satisfied all the prerequisites in [Using the SQL Converter](#).
- You have defined a token file to recognize the tokens. For more information see [Token File](#).

### Procedure

To convert SQL with tokens using the SQL converter, follow the same procedure as that in [Using the SQL Converter](#), except that, in addition, you must perform either of the following:

- Define the path of the token file in the configuration file. For more information, see [Configuration File](#).
- If you run the converter using the command line, define the path of the token file using the `t` parameter when you run the converter. For more information, see [Command-Line Parameters](#).

### Result

The defined SQL is converted from T-SQL grammar to ANSI-SQL grammar. You must perform checks on the converted version to ensure that it is correct according to your needs.



#### Example

The input file is as follows:

```
select Column1, [%1], Column2 from TABLE1 where Column1 = [%3];
```

After conversion, the output file is as follows:

```
SELECT "Column1", [%1], "Column2" FROM TABLE1 WHERE "Column1" = [%3];
```



#### Example

The input file is as follows:

```
SELECT $[OTABLE.U_FIELD] + $[$38.U_Numeric.Field] FROM OTABLE WHERE [LastName] Like  
'%[%0]%' and [Time] = [%1];
```

After conversion, the output file is as follows:

```
SELECT $[OTABLE.U_FIELD] + $[$38.U_Numeric.Field] FROM OTABLE WHERE "LastName" LIKE  
'%[%0]%' AND "Time" = [%1];
```

## 2.3.1 Token File

You use the token file to store the customized token patterns, so that the tokens in the SQL will be recognized. Often a column identifier is defined as a variable, which is replaced by a value used in a post-processing tool sending requests to the database server. To support such non-compliant SQL queries, the converter allows you to define regular expressions which correspond to standard SQL identifiers, and you can define as many patterns as you want.

In the downloaded SQL converter .zip file, the *tokenFile.txt* file contains the following patterns by default:

```
VarToken=\\$\\[\\$]*[0-9A-Z_a-z\\.\\"]+\\
```

```
VarToken=\\[\\%[0-9]+\\
```

```
IdToken=\\b[aA][sS]\\b[ ]+[gG][oO]\\b|\\b[aA][sS]\\b[\\n][gG][oO]\\b (This pattern represents a specific case where alias AS GO is supported in T-SQL while the GO statement represents a T-SQL keyword.)
```



### Example

The input file is as follows:

```
SELECT $[1.1] FROM OINV
```

The customized token *\$\$[1.1]* represents an identifier, or in a more general way, a variable. The predefined token pattern *VarToken=\\\$\\[\\\$]\*[0-9A-Z\_a-z\\.\\"]+\\* will be used to support this case.



### Example

The input file is as follows:

```
SELECT Name, Desc FROM [%4]
```

The customized token represents a table name, or in a more general way, an identifier. The predefined token pattern *IDToken=\\[\\%[0-9]+\\* will be used to support this case.

## 3 Using the SQL Converter via API

### Note

SQL conversion using API caters only to testing purposes during migration of software solutions, and not for real-time conversions in live systems because of its performance. The conversion process is fast without any significant delays, but the verification of the upper or lower-case letters in the names of database objects (e.g. columns, procedures, etc.) takes quite a lot of time.

You can also access the SQL converter function via API in your add-ons. In this way, you do not need to manually collect queries into the input file, and then run the command line. Instead, you can include the provided API in your add-on and convert your queries via the provided API while you use the add-on.

You can find examples (both 32-bit and 64-bit versions) of using the converter as a linked library in the converter folder. The examples are in the folder named [\[API\\_Example\]](#). For more information, see [API Example](#).

### Prerequisites

You have included the following libraries in your add-on:

- Antlr3.Runtime.dll
- ConverterLib.dll

You can find both libraries in the converter folder.

### Procedure

#### 1. Initializing the SQL converter.

The initialization of the converter is the initialization of the converter object. The initialization of the converter object is done through the object's public constructor. The construction requires the full file path of the configuration file to initialize the converter. A full file path means the file path and the file name. The configuration file is the same as previously stated in [Configuration File](#), except that the parameters for input and output files are not used.

#### Syntax

```
public TranslatorTool(string configPath)
```

#### Example

This example is in C#.

```
TranslatorTool tool = new TranslatorTool(@"..\..\Config.txt");
```

## 2. Converting the SQL statements.

The function `TranslatorTool::TranslateQuery` takes in TSQL strings and returns to the user converted strings in SQLScript.

### Syntax

```
string TranslatorTool::TranslateQuery(string inputQuery, out int numOfStatement,  
out int numOfErrors)
```

### Example

This example is in C#.

```
int stmts, errs;  
string result =  
tool.TranslateQuery("select isnull(code, 0) from oinv;", out stmts, out errs);
```

## 3. Closing the SQL converter.

The closing of the converter consists of the release of the converter object and the closing of allocated resources.

### Syntax

```
void TranslatorTool::Close()
```

### Example

This example is in C#.

```
tool.Close();
```

## 3.1 API Example

In the converter folder, you can find examples (both 32-bit and 64-bit versions) of using the converter as a linked library. The examples are in the folder named [\[API\\_Example\]](#). The folder contains a small Microsoft Visual Studio project with linked SQL converter libraries and examples. You can open the folder in Microsoft Visual Studio and directly run the prepared example.

## 4 Identifiers

In the T-SQL grammar, identifiers are not case-sensitive, but in the ANSI-SQL grammar, they are case-sensitive. When you are migrating SQL from the MS SQL Server database to the SAP HANA database, you need to fix the identifiers manually. The converter can verify and fix such identifiers with the *Use Case Fixer* option. When you use this option, identifiers in the SQL are fixed according to the existing database objects in the SAP HANA database, and comments also appear in the output file. For more information about this option, see [Using the SQL Converter](#).

### Note

When you use the *Use Case Fixer* option, you MUST define the connection details to the SAP HANA database.

### Example

The input file is as follows:

```
select docnum, DOCNUM, DocNum, nonExistingColumn from oinv;
```

After conversion, the output file is as follows:

```
SELECT "DocNum", "DocNum", "DocNum", "nonExistingColumn" FROM OINV;  
--[Note:CaseFixer] Column name docnum was changed to DocNum  
--[Note:CaseFixer] Column name DOCNUM was changed to DocNum  
--[Note:CaseFixer] Column nonExistingColumn not found in given tables  
--[Note:CaseFixer] Table name oinv was changed to OINV
```

In the SAP HANA database, non-quoted identifiers are considered as all-upper-case identifiers, and so to keep the upper or lower case of the identifiers, make sure to quote them. After conversion, all-upper-case identifiers are not quoted.

After conversion, the brackets of bracketed identifiers are removed, and the [dbo] identifier is removed.

### Example

The input file is as follows:

```
select MixedColumn, UCOLUMN, [BrackerColumn], "QuotedColumn", "UQUOTEDCOLUMN" from  
[dbo].TABLE1;
```

After conversion, the output file is as follows:

```
SELECT "MixedColumn", UCOLUMN, "BrackerColumn", "QuotedColumn", "UQUOTEDCOLUMN" FROM  
TABLE1;
```

## 5 Supported Data Types

The following table displays the data types supported by the SQL converter and their corresponding data types in the SAP HANA database.

### Supported Data Types

Data Type Category	Data Type in MS SQL Server Database	Data Type in SAP HANA Database
Exact numerics	bigint	bigint
	bit	tinyint
	decimal	decimal
	int	integer
	numeric	decimal
	smallint	smallint
	smallmoney	smalldecimal
	money	decimal
	tinyint	tinyint
Approximate numerics	float	float
	real	real
Date and time	date	date
	datetime2	timestamp
	datetime	timestamp
	datetimeoffset	SAP HANA does not support this data type.
	smalldatetime	seconddate
	time	time
Character strings	char	char
	varchar	varchar
	text	text
Unicode character strings	nchar	nchar
	nvarchar	nvarchar
	ntext	nclob



Data Type Category	Data Type in MS SQL Server Database	Data Type in SAP HANA Database
Binary strings	binary	binary
	varbinary	varbinary
	image	blob
Other Data Types	cursor	SAP HANA does not support this data type.
	timestamp	timestamp
	hierarchyid	SAP HANA does not support this data type.
	uniqueidentifier	nvarchar
	sql_variant	SAP HANA does not support this data type.
	xml	SAP HANA does not support this data type.
	table	SAP HANA does not support this data type.

## 6 Supported Operators

The following table displays the operators supported by the SQL converter.

### Supported Operators

Operator Category	Operator
Arithmetic operators	+ (add), - (subtract), * (multiply), / (divide)
String operators	(concatenation)
Comparison operators	=, >, <, >=, <=, <>, !=, !<, !>
Assignment operator	=
Bitwise operators	&,  , ^
Set operators	union, union all, intersect, except
Logical operators	and, or, not, all, any, between, exists, in, like, some, is null, is not null

---

## 7 Supported Expressions

SQL converter supports case expressions and aggregate expressions.

## 8 Supported SQL Statements

### 8.1 Supported DML

The following table displays the DML supported by the SQL converter, and related information.

DML	Related Information
Select	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms189499">http://msdn.microsoft.com/en-us/library/ms189499</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
Insert	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms174335">http://msdn.microsoft.com/en-us/library/ms174335</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
Update	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms177523">http://msdn.microsoft.com/en-us/library/ms177523</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
Delete	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms189835">http://msdn.microsoft.com/en-us/library/ms189835</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
WITH	In the MS SQL Server database: <a href="https://msdn.microsoft.com/en-us/library/ms175972">https://msdn.microsoft.com/en-us/library/ms175972</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .

### 8.2 Supported DDL

The following table displays the DDL supported by the SQL converter, and related information.

DDL	Related Information
Create Table	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms174979">http://msdn.microsoft.com/en-us/library/ms174979</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
Create Index	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms188783">http://msdn.microsoft.com/en-us/library/ms188783</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
Create View	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms187956">http://msdn.microsoft.com/en-us/library/ms187956</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .

DDL	Related Information
Alter Table	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms190273">http://msdn.microsoft.com/en-us/library/ms190273</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
Alter Index	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms188388">http://msdn.microsoft.com/en-us/library/ms188388</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
Drop Table	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms173790">http://msdn.microsoft.com/en-us/library/ms173790</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
Drop Index	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms176118">http://msdn.microsoft.com/en-us/library/ms176118</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .
Drop View	In the MS SQL Server database: <a href="http://msdn.microsoft.com/en-us/library/ms173492">http://msdn.microsoft.com/en-us/library/ms173492</a> In the SAP HANA database: look for the information in <i>SAP HANA SQL and System Views Reference</i> on SAP Help Portal at <a href="http://help.sap.com/hana_appliance">http://help.sap.com/hana_appliance</a> .

### 8.3 Supported Complex Statements

SAP HANA requires that you create stored procedures for cases of complex statements, that is, those statements with context dependencies. You can use the command-line parameter `P` to enable automatic creation of stored procedures in case of complex statements. If you do not use this parameter, the complex statements are converted to separate statements, that is, they are not wrapped into stored procedures.

The supported complex statements are: While statement, If statement, Declare Variable/Cursor, and Create/Drop/Alter procedure.

 Note

Variable is converted to plain text name.

## 9 Supported Features with Limitation

### 9.1 Concatenating Strings

In the T-SQL grammar, the operator "+" is used to concatenate the strings, while in the ANSI-SQL grammar, the operator "||" is used to concatenate the strings. In both grammars, the operator "+" is used in arithmetic addition.

The following table shows the rules that SQL converter follows for concatenating strings.

Rules for Concatenating Strings

Expression in Input File	Result in Output File
number + number	number + number
number + string	number + string
string + string	string    string
All other expressions	operand1 +operand2

### 9.2 Converting Date Formats

The T-SQL grammar contains several "date" formats, and the following sections describe how they are converted.

#### 9.2.1 Using String as DATE Parameter

In the T-SQL grammar, the commonly-used format "YYYYMMDD" is translated to "YYYY-MM-DD" as follows:

'20010101 10:00:00.000' => '2001-01-01 10:00:00.000'

All functions in SQL Script support the translated format.

 Note

All strings used as dates are trimmed from leading and trailing spaces as follows:

' 20010101 ' -> '2001-01-01'

---

## 9.2.2 Using Integer Value as DATE Parameter

In the T-SQL grammar, integer values are translated to DATE as follows:

0 = "1900-01-01"

1 = "1900-01-02"

In the ANSI-SQL grammar, there is no implicit translation from integer values to DATE. The SQL converter converts integer values used in the DATE argument of the procedure as follows:

Integer\_Value => ADD\_DAYS("1900-01-01 00:00:00.000, Integer\_Value)

[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2016 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System ads, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, ISeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, R/3, xApps, xApp, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.