12.12.2019

# LCM-331 Feld lcm\_document\_link

Alle Dokumente im Zusammenhang mit Cases-Communication sind in BSI\_DOCUMENT.CONTENTS gespeichert. Der Inhalt dieses Feldes ist verschlüsselt. LCM enthält Methoden, den Inhalt zu entschlüsseln, in ein File zu schreiben und dieses im Benutzer-Verzeichnis zu speichern. Von dort wird es per Batch Befehl geöffnet.

Für den Case-Export werden In lcm\_document\_link aktuell alle Dokument-Namen als Komma-separierter String zu einem Wert von BSI\_COMMUNICATION.COMMUNICATION\_NR gespeichert; d.h. BSI\_COMMUNICATION und BSI\_DOCUMENT haben eine 1:n Beziehung. Damit Dokumente in salesforce angezeigt werden können, müssen zusätzlich BSI\_DOCUMENT.DOCUMENT\_NR und BSI\_DOCUMENT.CONTENT exportiert werden. Probleme macht das Datenvolumen: für Dokumente der letzten 2 Jahre (2018, 2019) beträgt es 80 GB. Da die Daten in der Cloud gespeichert werden, ist das ein Kostenfaktor.

## Besprechung vom 16.12.2019 mit Dincer, Urs, Röbi, ich

Die Daten aus BSI\_DOCUMENT.CONTENT zusammen mit notwendigen Schlüsselfeldern wie BSI\_DOCUMENT.DOCUMENT\_NR, BSI\_CASE.CASE\_NR, BSI\_COMMUNICATION.COMMUNICATION\_NR und evt. weitere werden im Datalake gespeichert. Auf Salesforce-Seite muss ein Mechanismus analog dem in LCM erstellt/konfiguriert werden, welcher die die CONTENT-Daten anhand ihres Schlüssels aus dem Datalake abfragt, in ein File schreibt und das File in das User-Directory schreibt.

12.11.2019

# LCM-331 Cases selektieren

Daten aus BSI\_CASE und BSI\_COMMUNICATION sollen selektiert und für den csv Export in ein File geschrieben werden.

## BSI\_COMMUNICATION.TEXT\_LOB: Probleme beim Extrahieren eines Teilstrings

Die ersten 4000 Zeichen des CLOB-Feldes BSI\_COMMUNICATION.TEXT\_LOB sollen extrahiert werden. Dazu kann die Oracle Funktion:

dbms\_lob.**substr**(TEXT\_LOB, [Anzahl\_Zeichen], [Start\_Position])

mit Anzahl\_Zeichen = 4000 und Start\_Position = 1 verwendet werden. Beim Auslesen bestimmter CLOB-Daten wird dabei der Fehler

ORA-06502: PL/SQL: numeric or value error: character string buffer too small

erzeugt.

**Grund:** Multibyte-Zeichen benötigen 2 bis 4 Byte.

**Lösung:** Erstellen von 4 Teilstrings zu je 1000 Zeichen.

**CASE** **WHEN** CMC.TEXT\_LOB **IS** **NOT** **NULL** **THEN**

**CASE** **WHEN** DBMS\_LOB.GETLENGTH(CMC.TEXT\_LOB) > 4000 **THEN**

'(4000 characters from total ' || DBMS\_LOB.GETLENGTH(CMC.TEXT\_LOB) || ')' || **chr**(13)

|| **REPLACE**(**REPLACE**(dbms\_lob.**substr**(CMC.TEXT\_LOB, 1000, 1), **CHR**(13), ' '), **chr**(11), ' ')

**ELSE**

'(' || DBMS\_LOB.GETLENGTH(CMC.TEXT\_LOB) || ' characters)' ||

**REPLACE**(**REPLACE**(dbms\_lob.**substr**(CMC.TEXT\_LOB, 1000, 1), **CHR**(13), ' '), **chr**(11), ' ')

**END**

**ELSE**

**NULL**

**END** LCM\_Communcation\_Text\_Part1,

**CASE** **WHEN** CMC.TEXT\_LOB **IS** **NOT** **NULL** **AND** DBMS\_LOB.GETLENGTH(CMC.TEXT\_LOB) > 1000 **THEN**

**REPLACE**(**REPLACE**(dbms\_lob.**substr**(CMC.TEXT\_LOB, 1000, 1001), **CHR**(13), ' '), **chr**(11), ' ')

**ELSE**

**NULL**

**END** LCM\_Communcation\_Text\_Part2,

**CASE** **WHEN** CMC.TEXT\_LOB **IS** **NOT** **NULL** **AND** DBMS\_LOB.GETLENGTH(CMC.TEXT\_LOB) > 2000 **THEN**

**REPLACE**(**REPLACE**(dbms\_lob.**substr**(CMC.TEXT\_LOB, 1000, 2001), **CHR**(13), ' '), **chr**(11), ' ')

**ELSE**

**NULL**

**END** LCM\_Communcation\_Text\_Part3,

**CASE** **WHEN** CMC.TEXT\_LOB **IS** **NOT** **NULL** **AND** DBMS\_LOB.GETLENGTH(CMC.TEXT\_LOB) > 3000 **THEN**

**REPLACE**(**REPLACE**(dbms\_lob.**substr**(CMC.TEXT\_LOB, 1000, 3001), **CHR**(13), ' '), **chr**(11), ' ')

**ELSE**

**NULL**

**END** LCM\_Communcation\_Text\_Part4,

Die 4 Teilstrings werden für die Ausgabe in das csv File konkateniert:

|| **REPLACE**(**REPLACE**(r.LCM\_Communcation\_Text\_Part1,'"',''''),'''''','''')

|| **REPLACE**(**REPLACE**(r.LCM\_Communcation\_Text\_Part2,'"',''''),'''''','''')

|| **REPLACE**(**REPLACE**(r.LCM\_Communcation\_Text\_Part3,'"',''''),'''''','''')

|| **REPLACE**(**REPLACE**(r.LCM\_Communcation\_Text\_Part4,'"',''''),'''''','''')

|| '","' || LCM\_Document\_link

Die Funktion dbms\_lob.**substr** gibt **varchar2** zurück. **varchar2** kann seit Oracle Version 12.1 bis zu 32767 Bytes aufnehmen, also 8191 Zeichen a 4 Byte. Nach der alten Definition kann **varchar2** aber nur 4000 Bytes aufnehmen. Mit einer Begrenzung von 1000 Zeichen für Anzahl\_Zeichen ist man dann auf der sicheren Seite.