8．16工作汇报

**一．遗留问题说明**

**1．关于LOB列不能做主键或主键的一部分**

原因：当lob列作为主键时，如果更新表中的其它列，则在更新日志中由于lob列没有更新，所以无法捕获它的值，xstream获得的日志信息中lob列的值用unavialible填充。此时在onlineMigration端生成的插入或更新语句由于没有主键，则无法插入到openGauss数据库中。

有如下表：

|  |  |
| --- | --- |
| Name | Type |
| ID | NUMBER(38) |
| J1 | VARCHAR2(255) |
| J2 | CLOB |

更改J2列：

"payload": {

    "before": {

        "ID": "Ag==",

        "J1": "{\"json1\":\"nothing\"}",

        "J2": "\_\_debezium\_unavailable\_value"

    },

    "after": {

        "ID": "Ag==",

        "J1": "{\"json1\":\"nothing\"}",

        "J2": "{\r\n  \"name\": \"clob\",\r\n  \"version\": \"0.0.1\",\r\n  \"lockfileVersion\": 1,\r\n  \"requires\": true\r\n}"

    }

}

更改J1列：

"payload": {

    "before": {

        "ID": "Ag==",

        "J1": "{\"json1\":\"nothing\"}",

        "J2": "\_\_debezium\_unavailable\_value"

    },

    "after": {

        "ID": "Ag==",

        "J1": "{\"json1\":\"something\"}",

        "J2": "\_\_debezium\_unavailable\_value"

    }

}

**2．对象类型中不能有数据组类型**

原因：在oracle向openGauss数据库类型转化方案中确定object类型在openGauss中转换成json格式，示例如下：

在 Oracle 中定义 TestType：

CREATE OR UPDATE TYPE TestType AS OBJECT (

id INTEGER,

foo VARCHAR(255)

);

若赋值 id=5, foo="wo"，则对应JSON为：

{

"id": 5,

"foo": "wo"

}

当对象类型的数据中嵌套数组类型的数据时，其定义如下：

create type debezium.arraychar as varray(255) of varchar2(255);

/

create type debezium.arrayarray as varray(255) of arraychar;

/

create table testarrayarray(

id int primary key,

name varchar(255),

a arrayarray);

create type debezium.arrayobj as object (

id int,

a arrayarray);

/

create table testobjarray(

id int primary key,

name varchar(255),

o arrayobj);

insert into testobjarray values(5,'eee',

arrayobj(44,

arrayarray(arraychar('a'),arraychar('a','b'),arraychar('a','bb','ccc'))

)

);

xstream获得的日志信息是以xml的形式表达的，如下：

<?oracle-xmldoc versions="D0 C0 V0" ?>

<ARRAYOBJECT>

    <ID>44</ID>

    <A>

        <ARRAYCHAR>

            <VARCHAR2>a</VARCHAR2>

        </ARRAYCHAR>

        <ARRAYCHAR>

            <VARCHAR2>a</VARCHAR2>

            <VARCHAR2>b</VARCHAR2>

        </ARRAYCHAR>

        <ARRAYCHAR>

            <VARCHAR2>a</VARCHAR2>

            <VARCHAR2>bb</VARCHAR2>

            <VARCHAR2>ccc</VARCHAR2>

        </ARRAYCHAR>

    </A>

</ARRAYOBJECT>

如果数据库中某一行数据对于对象类型的列赋值时，数组只赋予了一个元素，则其在xml的表达形式下是无法确定该元素是数组还是其它类型，因此会造成数据解析上的错误，从而使得转换成json时产生错误。如下所示：

上述插入语句预期结果：

{"A":[["a"],["a","b"],["a","bb","ccc"]],"ID":44}

实际结果：

{"A":[{"VARCHAR2":"a"},["a","b"],["a","bb","ccc"]],"ID":44}

原因在于XML中ARRAYOBJECT/A/ARRAYCHAR下仅包含一个节点，无法判断该类型是数组。

再比如更换数组类型如下

object类型testtype：

create type testtype as object (

id int,

str varchar(255)

);

数组类型arrayobj：

create type arrayobj as varray(5) of testtype;

表定义：

create table testobjarray(

id int primary key,

a arrayobj

);

插入数据后可得到如下XML：

<?oracle-xmldoc versions="D0 C0 V0" ?>

<ARRAYOBJECT>

    <ID>44</ID>

    <A>

        <TESTTYPE>

            <ID>1</ID>

            <STR>sth</STR>

        </TESTTYPE>

    </A>

</ARRAYOBJECT>

此时的预期结果是：

{"A":[{"ID": 1, "STR": "sth"}],"ID":44}

而实际结果是

{"A":{"TESTTYPE": {"ID": 1, "STR": "sth"}},"ID":44}

因此建议oracle数据库中定义的object类型中不要嵌套数组类型。

**3．关于oracle中long类型数据最大容量的问题**

在Oracle官方文档中LONG类型容量上限为2GB-1字节，为了验证是否属实进行了插入和查询实验。使用字符串或文件形式插入较长文本会超出sqlplus的长度限制或oracle的缓冲区限制，所以尝试通过SQLDeveloper、Navicat、DBeaver等方法以文件的方式插入数据，如图1所示。



图1. 1.9GB文本测试文件

实验时我们采用逐渐增加数据量的实验方法，从32M，64M，128M、512M…，直至1.9GB这种增量的方式进行实验。

实验过程中，为确定插入数据是成功的，我们将插入的数据导出后进行验证。但发现当插入的数据较大时，导出的数据被截断不完整，比如插入1.9G的数据后(如图2所示)，通过DBeaver导出时报错“java.lang.OutOfMemoryError: Java heap space”(如图3所示)，怀疑内存不足导致，但该工具无法进行内存容量配置，所以暂时无法验证插入的测试数据是否完整。

但目前在通过Navicate导入的功能插入1.9GB数据时显示成功执行，且DBeaver显示在插入前TEST\_LONG表占用64KB、插入后TEST\_LONG表占用空间增加至2GB(如图4所示)，猜测已成功插入1.9GB的文本数据，但暂时无法查看插入的内容是否与原文件一致、完整。

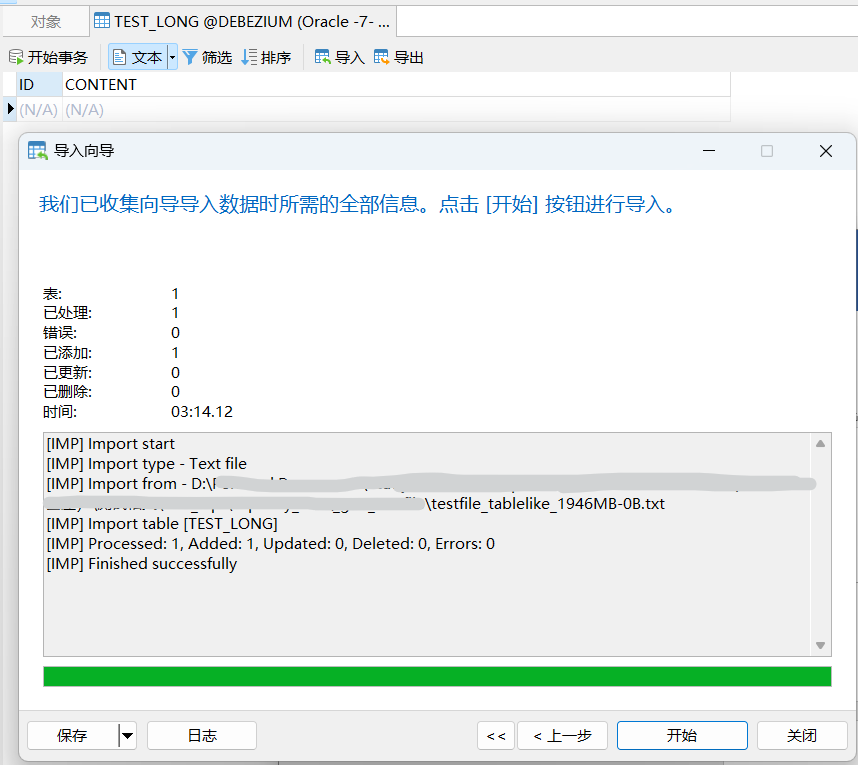


图2. 成功通过Navicat插入1.9GB的数据

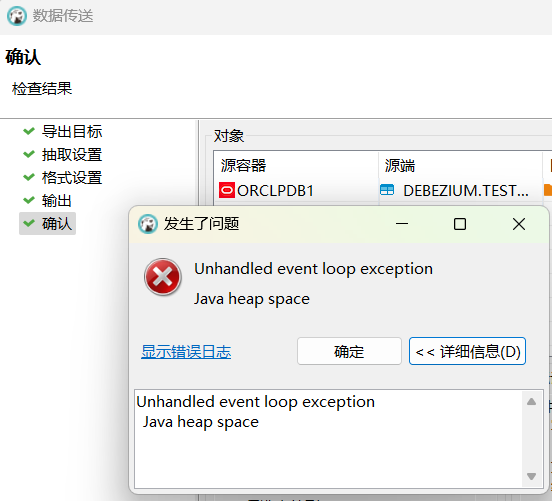


图3. DBeaver导出长数据时报错



插入前-表占用容量（64KB）



图4. 插入后-表占用容量（2GB，接近1.9GB）

**二．下一步的工作**

下一步再以long类型为例测试xstream捕获日志中数据的大小以及debezium

杜金莲

北京工业大学计算机学院

2022.08.17