Database - Bug #5013

Collision of java names with SQL keywords

11/20/2020 05:16 PM - Ovidiu Maxiniuc

Status:	Test	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Ovidiu Maxiniuc	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**case_num:**	
vendor_id:	GCD		

Description

History

#1 - 11/20/2020 05:22 PM - Ovidiu Maxiniuc

I encountered the following issue.

There is a (database) table named order. The conversion correctly identifies the possible issue and generates the SQL name order_. The ABL source is:

```
FOR EACH order:
    DELETE order.
END.
```

The converted code looks like this:

```
    forEach(TransactionType.FULL, "loopLabel2", new Block((Init) () ->
    {
       query2.initialize(order, ((String) null), null, "order.ordernum asc");
    },
```

The problems start when attempting to convert the following FQL:

```
    select
        order.recid
    from
        Order__Impl__ as order
    order by
        order.ordernum asc
```

As you notice, there are two consecutive order tokens! Somehow, the parser stops at as so only the following tree is generated:

```
    select statement [SELECT_STMT]:null @0:0
       select [SELECT]:null @1:1
          order [ALIAS]:null @1:8
             recid [PROPERTY]:null @0:0
       from [FROM]:null @1:20
          Order__Impl__ [DMO]:null @1:25
```

There is no parsing warning, this way it managed to fly *under the radar* and there are no issues reported to log.

I think it is difficult to predict these kind of (more or less correct) FQL statements. This is just an example, but think what the statement would look if the table's Java name was select, limit, delete, etc. A way to handle this is using table hints, but this implies we know the issue a-priori. A better solution would be FWD to handle it automatically. I wonder whether if we could avoid such issues by detecting this kind of collisions at conversion time and use an alternative Java name for these buffer variables?

#2 - 11/23/2020 10:51 AM - Ovidiu Maxiniuc

- Status changed from New to WIP

I have fixed the issue by filtering through NameConverter.resolvePossibleKeywordConflict() the result of P2OAccessWorker.javaBufferName() and P2OAccessWorker.javaPropertyName() before returning them. As result, the order or by buffer names are now better converted to order_ and by_ Java identifier/DMO alias. The same result will be obtained if the code encounters fields (properties) with same name.

A bit off topic, I encountered another issue. Suppose we have a procedure that dynamically handles tables. Its first two statement are:

```
DEFINE VARIABLE bh AS HANDLE.
CREATE BUFFER bh FOR TABLE "order".
```

Since this is really the first occurrence of order table, and there are no static usage of the table/buffer, the DMO was not processed. Now, we might attempt to force constructing the DMO path and loading it in CREATE BUFFER implementation using DmoMetadataManager.getDmoBasePackage() + "." + databaseName + "." + tableName. The problem is, we do not really know the DMO tableName, only the legacy. And, as noted above, it might not be straightforward association. Beside adding the underscore (as NameConverter.resolvePossibleKeywordConflict() does), the DMO name might have been manually set using conversion hints.
I am a bit blocked here.

#3 - 12/07/2020 03:35 PM - Ovidiu Maxiniuc

This issue should be fixed in r11866/4397a (11/26/20).

#4 - 12/07/2020 03:36 PM - Ovidiu Maxiniuc

- % Done changed from 0 to 100

#5 - 12/07/2020 04:58 PM - Greg Shah

- Assignee set to Ovidiu Maxiniuc

- Status changed from WIP to Test