Base Language - Bug #6412

implement BUFFER:KEYS attribute

05/26/2022 06:04 AM - Constantin Asofiei

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

Description

History

#2 - 05/26/2022 06:05 AM - Constantin Asofiei

There is this abend as BufferImpl does not implement 'Keyable' methods:

```
java.lang.AbstractMethodError: Method com/goldencode/p2j/persist/$__Proxy1226.keys()Lcom/goldencode/p2j/util/c
haracter; is abstract
        at com.goldencode.p2j.persist.$__Proxy1226.keys(Unknown Source)
```

#3 - 05/26/2022 06:27 AM - Constantin Asofiei

These are all the methods not implemented in BufferImpl:

- The type BufferImpl must implement the inherited abstract method Keyable.keys(long)
- The type BufferImpl must implement the inherited abstract method Keyable.keys(NumberType)
- The type BufferImpl must implement the inherited abstract method Buffer.querySynchronize()
- The type BufferImpl must implement the inherited abstract method IterableResource.setCurrentIteration(handle)
- The type BufferImpl must implement the inherited abstract method Keyable.keys()

#4 - 05/26/2022 03:52 PM - Ovidiu Maxiniuc

- % Done changed from 0 to 50

- Status changed from New to WIP

I am about to finish the implementation of the missing methods.

#5 - 05/26/2022 07:43 PM - Ovidiu Maxiniuc

- Status changed from WIP to Review

- % Done changed from 50 to 100

The implementation of the above methods were committed to 6129a as r13892.
Some notes about the revision:

- CURRENT-ITERATION returns an INTEGER for BUFFER not a HANDLE. I do not know how we can detect and resolve this at conversion time if all we have is a HANDLE. It is a read-only attribute and I implemented the setter. However, the name of the setter must be changed to currentIteration() instead of setCurrentIteration(). I cannot return an integer at this time.
- I had some difficulties working with buffers. In fact this was my great time consumer. The fix was simple, just inserting the .ref(). The problem was that the debugger showed me the object structure correct before calling a method but when the method was invoked (using proxy), the object was 'garbaged';
- there was a problem handling the buffers lists in dynamic query handler. Because they were added twice, a circular loop was created leading the application freeze when this tree was navigated;
- I noticed that the static data-source's name was not initialized at all. Fixed it.

#6 - 05/27/2022 01:40 AM - Constantin Asofiei

Ovidiu, I don't think the implemention is correct.

> If there are no defined key fields, this attribute returns a comma-separated list of key fields in the buffer’s unique primary index (if any). If there are no defined key fields and no unique primary index, this attribute returns the string "ROWID".

There is application logic which looks into the KEYS fields, and if there is none, you return ROWID... there is no such field.

#7 - 05/27/2022 01:18 PM - Ovidiu Maxiniuc

> Ovidiu, I don't think the implemention is correct.

The implementation is based on testcases. Like this:

```
DEFINE TEMP-TABLE t3t FIELD f2f AS INTEGER.
MESSAGE BUFFER t3t:KEYS.
```

will print rowid (in lowercase).

However, if you construct a buffer with rowid as keys in a dataset, the method will return ROWID, in uppercase. Strange.

#8 - 05/27/2022 01:30 PM - Constantin Asofiei

Add a unique, primary (and combinations of unique or not, primary or not) index and see what happens. My patch is this:

```
### Eclipse Workspace Patch 1.0
#P p2j6129a
Index: src/com/goldencode/p2j/persist/BufferImpl.java
===================================================================
--- src/com/goldencode/p2j/persist/BufferImpl.java    (revision 3634)
+++ src/com/goldencode/p2j/persist/BufferImpl.java    (working copy)
@@ -4641,6 +4641,31 @@
    @Override
    public character keys()
    {
+      if (parentDataSource == null || keys == null)
+      {
+         // get the primary unique index
+         int idxNo = 0;
+         TableMapper.LegacyIndexInfo index = TableMapper.getLegacyIndexInfo(buffer(), idxNo);
+         while (index != null)
+         {
+            if (index.isUnique() && index.isEffectivePrimary())
+            {
+               String bufKeys = "";
+               for (String key : index.getComponents())
+               {
+                  if (!bufKeys.isEmpty())
+                  {
+                     bufKeys += ",";
+                  }
+                  bufKeys += key;
+               }
+
+               return new character(bufKeys);
+            }
+            index = TableMapper.getLegacyIndexInfo(buffer(), ++idxNo);
+         }
+      }
+
       if (parentDataSource == null)
       {
          return new character(DataSource.ROWID_KEY);
@@ -4654,6 +4679,11 @@
       StringBuilder sb = new StringBuilder();
       for (String key : keys)
       {
+         if (Session.PK.equals(key))
+         {
+            continue;
+         }
+
          if (sb.length() != 0)
          {
             sb.append(",");
```

#9 - 05/27/2022 02:52 PM - Ovidiu Maxiniuc

I see. The implementation was incomplete. Thank you for noticing that.
I used your patch to fix the issue. Notice that using the TableMapper implies a double map lookup. The preferred approach is to access the DMO metadata directly.
Committed as r13894.

#10 - 06/01/2022 04:36 PM - Constantin Asofiei

- Status changed from Review to Test