

## Database - Bug #2526

### Records preserving its position in the sorting index for adaptive query

02/27/2015 08:17 AM - Stanislav Lomany

<b>Status:</b>	New	<b>Start date:</b>	02/27/2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Cleanup and Stabilization for GUI	<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

### History

#### #1 - 02/27/2015 08:28 AM - Stanislav Lomany

- Target version set to Milestone 12

This problem was noticed in an editable browse which easily exposes this bug. Assigning a value to a field of the index which drives an adaptive query for a browse discards not yet loaded records into browse which are now positioned before the updated record.

Testcase 1 (scrolling query):

```
def temp-table tt field f1 as integer
    index idx f1.
```

```
def var i as integer.
repeat i = 1 to 10:
    create tt. tt.f1 = i.
end.
```

```
def query q for tt scrolling.
open query q for each tt.
```

```
form tt.f1 with frame f1 15 down.
```

```
repeat:
    get next q.
    if not avail(tt) then leave.
```

```
if tt.f1 = 5 then do transaction:
    tt.f1 = 9999.
    display tt.f1 with frame f1.
    down with frame f1.
```

```
reposition q to row 3.
```

```
end.
else do:
    display tt.f1 with frame f1.
    down with frame f1.
end.
```

```
end.
```

4GL output:

```
| f1 |
|----|
| 1  |
```

2
3
4
9,999
3
4
9,999
6
7
8
9
10
9,999

P2J output:

f1
1
2
3
4
9,999
3
4
9,999

Testcase 2 (non-scrolling query):

```
def temp-table tt field f1 as integer
    index idx f1.
```

```
def var i as integer.
repeat i = 1 to 10:
    create tt. tt.f1 = i.
end.
```

```
def query q for tt.
open query q for each tt.
```

```
form tt.f1 with frame f1 15 down.
```

```
repeat:
    get next q.
    if not avail(tt) then leave.
```

```
    if tt.f1 = 5 then do transaction:
        tt.f1 = 9999.
        display tt.f1 with frame f1.
        down with frame f1.
```

```
        get current q.
    end.
    else do:
        display tt.f1 with frame f1.
        down with frame f1.
    end.
```

```
end.
```

4GL output:

f1
1
2
3
4
9,999
6
7
8
9
10
9,999

P2J output:

f1
1
2
3
4
9,999

**#2 - 02/27/2015 08:37 AM - Stanislav Lomany**

As for initial analysis: currently we use record snapshots as the base to find the next record in the index using an HQL statement. So we can change an indexed field and properly found the next record. But it looks like we should preserve these snapshots for longer time:

- for non-scrolling queries the snapshot should survive record reload;
- for scrolling queries snapshot should be preserved for the last record cached by the cursor.

**#3 - 03/23/2016 05:19 PM - Greg Shah**

- *Target version changed from Milestone 12 to Milestone 16*

**#4 - 11/16/2016 12:23 PM - Greg Shah**

- *Target version changed from Milestone 16 to Cleanup and Stabilization for GUI*