

Database - Bug #11846

Inconsistent session cache when evicting last record of a ProgressiveResults bracket

09/08/2026 06:29 AM - Alexandru Lungu

Status:	Review	Start date:	
Priority:	Normal	Due date:	
Assignee:	Alexandru Lungu	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		reviewer:	Constantin Asofiei
billable:	No	production:	No
vendor_id:	GCD	env_name:	
case_num:		topics:	
version_reported:			
version_resolved:			
Description			

History

#2 - 09/08/2026 06:59 AM - Alexandru Lungu

- % Done changed from 0 to 100
- Assignee set to Alexandru Lungu
- Status changed from New to Review
- reviewer Constantin Asofiei added

I discovered this bug in my tests with two sessions (one reading and the other one writing) in a stressful manner. In ProgressiveResults.getResults:

```
if (delegate != null && delegate.last())
{
    lastRecord = delegate.get();
}
```

lastRecord is a reference to a record that is cached.

After this, the delegate is closed and a new bracket is retrieved and eventually delegate.first() is called. There is a chance that this delegate.first() to require space in the Session.cache and actually evict lastRecord which is not pinned by any buffer.

Later on:

```
bufMgr.evictDMOIfUnused(persistence, null, dmo);
```

Evicting the dmo (which is computed from lastRecord) yields Inconsistent session cache log because we try to evict a DMO that is no longer cached. This is not fatal, but may trigger a severe log.

The real functional risk is that the eviction is triggered under !dmo.isInUse(), but this dmo **may not be cached**, so we end up running session.evict(dmo); and evict the **cached** image which **might still be referenced**.

I fixed this in 11846a/rev. 16746

Constantin, please review.