

EDIT - bug #2851

Check ordering in API paged methods

03/13/2012 12:16 PM - Andreas Müller

Status:	Discussed	Start date:	
Priority:	Highest	Due date:	
Assignee:	Andreas Müller	% Done:	0%
Category:	cdmlib	Estimated time:	0:00 hour
Target version:	Release 5.45	Found in Version:	
Severity:	normal		
Description			
Paging API method call results may cause problems if the result is not ordered. We need to check our methods and fix them if needed.			
Related issues:			
Related to EDIT - bug #9948: Search result contains only 24 items and the nex...		Closed	
Related to EDIT - bug #9943: webservice portal/agent delivers wrong page content		Closed	

History

#1 - 03/13/2012 12:17 PM - Andreas Müller

Stimmt, da haste Recht! Ich hatte lediglich an den recht einfachen Fall `select * from gedacht`.

Eine Lösung wäre bei allen toMany Relations `@OrderBy(titleCache)` oder `@OrderBy(id)` hinzu zufügen.

Andreas

---Ursprüngliche Nachricht---

Von: Müller, Andreas

Gesendet: Freitag, 9. März 2012 11:41

An: Kohlbecker, Andreas

Betreff: AW: Ordering in paged methods

Hi Andreas,

Dass sehe ich nicht so und habe es auch schon erlebt. Zwar stimmt es in aller Regel, dass das DBMS bei 2 gleichen Anfragen diese in der gleichen Reihenfolge zurückgibt, aber wenn kein Order BY angegeben ist, wird dir kein DBMS die Garantie geben, dass das so sein muss.

Da sortieren ein sehr teurer Prozess, versucht das DBMS zu umgehen, sich um Sortierung kümmern zu müssen und produziert die Daten in einfachsten Art und Weise. Wenn du z.B. ein WHERE statement hast mit `WHERE id IN (a,b,c)` kommt i.d.R. etwas anderes heraus als `WHERE id IN (c,a,b)`. D.h. hier wird sich nicht am PK orientiert, während das in anderen Fällen der Fall ist. Auch kann LIMIT 100000 zu anderen Ergebnissen führen als LIMIT 10 da der Optimierer vielleicht einen anderen Execution Plan wählt und somit die Datenstrom Pipeline zuerst andere Daten ausspuckt.

In den allermeisten Fällen und insbesondere beim Paging über gleiche Partitionsgrößen und quasi unveränderte Daten hast du zwar sicherlich Recht, aber sicherer fände ich es schon, wenn wir ein festdefiniertes Verhalten vorgeben.

Können wir ja mal mündlich besprechen.

AM

---Ursprüngliche Nachricht---

Von: Kohlbecker, Andreas

Gesendet: Freitag, 9. März 2012 10:25

An: Müller, Andreas

Betreff: AW: Ordering in paged methods

Hallo Andreas,

ich nehme an solange kein explizites "order by" in hql oder als criteria angegeben ist wird hibernate in das sql statement auch kein order by einbauen. Das heißt es steht im diesem Fall der Datenbank frei zu sortieren wie diese gerne möchte. [MySQL](#) sortiert per default nach dem PK. Alle Datenbanken haben da mit Sicherheit irgendeine Defaultsortierreihenfolge, ansonsten müsste man immer wenn man LIMIT oder ähnliches verwendet gezwungenermaßen ein order by angeben, was ja nicht der Fall ist. Service seitig sind wir damit auf der sicheren Seite was das Paging betrifft.

Andreas

----Ursprüngliche Nachricht-----

Von: Müller, Andreas

Gesendet: Montag, 5. März 2012 14:46

An: Kohlbecker, Andreas

Betreff: Ordering in paged methods

Hi Andreas,

Da du wohl krank bist, diese Anfrage per Mail:

Ich habe festgestellt, dass es in der Service Layer öfters Methoden gibt die zwar Partitionen oder Pages zurückgeben, aber keine Hinweise zum Ordering enthalten. Das verstehe ich nicht ganz, da ein Paging doch nur möglich ist, wenn ich eine definierte Reihenfolge habe.

Z.B: wie garantiere ich in `DescriptionDaoImpl.getTaxonDescriptions(xxx)`, page 2 die Records zurückgibt, die nach Page 1 folgen. Sortiert Hibernate automatisch nach id, wenn es sonst kein Kriterium gibt?

Dies ist dringend für mich, da wir das im PESI Export regelmäßig über die Gesamtheit aller Partitionen iterieren und ich sicherstellen muss, dass damit auch wirklich alle Records abgedeckt sind und gleichzeitig keine Duplikate entstehen.

Viele Grüße,

Andreas M.

#2 - 11/23/2021 07:35 PM - Andreas Müller

- *Description updated*

#3 - 02/22/2022 07:58 PM - Andreas Müller

- *Status changed from New to Discussed*

- *Priority changed from Priority13 to Highest*

- *Target version changed from cdmlib - Old Next Major Release to Release 5.45*

If this is still an open issue it is urgent.

#4 - 02/22/2022 07:58 PM - Andreas Müller

- *Private changed from Yes to No*

#5 - 02/22/2022 07:59 PM - Andreas Müller

- *Related to bug #9948: Search result contains only 24 items and the next page starts with the 26th result added*

#6 - 02/22/2022 08:03 PM - Andreas Müller

- *Related to bug #9943: webservice portal/agent delivers wrong page content added*

#7 - 02/22/2022 08:04 PM - Andreas Müller

- *Tags set to 5.30*

#8 - 03/03/2022 11:02 AM - Andreas Müller

- *Tags deleted (5.30)*