

Database - Feature #1583

add conversion and runtime support for the INSERT language statement

10/11/2012 11:19 PM - Eric Faulhaber

Status:	Closed	Start date:	11/28/2012
Priority:	Normal	Due date:	
Assignee:	Ovidiu Maxiniuc	% Done:	100%
Category:		Estimated time:	8.00 hours
Target version:	Conversion Support for Server Features	vendor_id:	GCD
billable:	No		
Description			

History

#1 - 10/22/2012 02:14 PM - Eric Faulhaber

- Start date deleted (10/15/2012)

#2 - 10/31/2012 04:13 PM - Greg Shah

- Target version set to Milestone 4

#3 - 11/22/2012 12:17 AM - Eric Faulhaber

- Assignee set to Ovidiu Maxiniuc

Greg,

There are only 4 uses of this statement across both projects (all in the server project), and they are all of the simplest possible form: INSERT <table> (i.e., no options). I would propose that our initial implementation is simply to convert:

```
INSERT <table>.
```

to

```
CREATE <table>.  
UPDATE <table>.
```

during annotations (or perhaps even earlier...fixups?) -- *provided testing in the 4GL shows these are equivalent* -- and let the normal conversion run its course.

We can create a separate issue to make a prettier and/or more compact conversion and to support the various options of INSERT later if necessary. In fact, as part of that, we could recognize the CREATE, UPDATE sequence and refactor that into the better conversion if the use of options (or lack thereof) allow it.

Thoughts?

#4 - 11/22/2012 09:17 AM - Greg Shah

Yes, I agree completely. So far we have almost never seen this statement used in practice. It is an interesting comment on the value of it. In 15 million lines of code (and probably thousands of CREATES), there are only 4 uses of INSERT.

Let's support it (using your approach) but there is no reason to anything more.

#5 - 11/28/2012 04:15 AM - Ovidiu Maxiniuc

- File *om_upd20121128a.zip* added

I first tested if the sequence of CREATE/UPDATE has the same effect like INSERT. My fear was that if user skips the UPDATE screen a new record would be inserted into table, which should not happen in the case of INSERT as this would be an atomic change to table. First, the real thing, on windev01. A simple program like

```
CREATE Customer.  
UPDATE Customer.
```

and skipping in update, strangely, did not create any new record into table, as the new record is created but not committed. I ran the same small program on P2J and.. surprise, the same result.

The implementation. In *post_parse_fixups.xml*, when walking on a STATEMENT which has only one child of type INSERT, it is replaced by two duplicates, but having their only child changed to CREATE / UPDATE. This way, the newly added nodes keep lexer information about line/col of the original INSERT statement node.

I uploaded the update which contains a testcase, too, please review.

#6 - 11/28/2012 04:18 AM - Ovidiu Maxiniuc

- Status changed from *New* to *WIP*

#7 - 11/28/2012 04:18 AM - Ovidiu Maxiniuc

- Status changed from *WIP* to *Review*

#8 - 12/07/2012 01:54 PM - Eric Faulhaber

- % Done changed from 0 to 70

- Start date set to 11/28/2012

Code looks good, but I should not have suggested that this go into post-parse fixups. This is too early, and it will result in the analysis reports (which run after post-parse fixups) being misleading, since they will not show any INSERT statement use.

It is better placed in the annotations phase. Specifically, the insert expansion rule must take place before *annotations/record_field_expansion.rules*. Please create *annotations/insert_statement_expansion.rules* and call it from within *annotations.xml*, right at the beginning of the database support section, just before *record_field_expansion*.

#9 - 01/10/2013 09:26 AM - Ovidiu Maxiniuc

- File *om_upd20130110b.zip* added

Replacement of INSERT by CREATE+UPDATE moved to new ruleset file. The intermediary files generated differ just by peerids.

#10 - 01/31/2013 11:09 PM - Eric Faulhaber

- % Done changed from 70 to 100

- Status changed from Review to Closed

#11 - 11/16/2016 11:08 AM - Greg Shah

- Target version changed from Milestone 4 to Conversion Support for Server Features

Files

om_upd20121128a.zip	10.6 KB	11/28/2012	Ovidiu Maxiniuc
om_upd20130110b.zip	14.2 KB	01/10/2013	Ovidiu Maxiniuc