Core Development - Feature #1468

finish the query_sorting.odt chapter of the P2J Conversion Reference

08/06/2012 02:23 AM - Eric Faulhaber

Status:	Review		**Start date:**	08/06/2012
Priority:	Normal		**Due date:**	
Assignee:	Adrian Lungu		**% Done:**	90%
Category:			**Estimated time:**	0.00 hour
Target version:				
billable:	No		**version:**	
vendor_id:	GCD			

Description

Related issues:		
Blocks Core Development - Feature #1469: slides for the "Query Results Sortin...	**Review**	**08/06/2012**

History

#1 - 08/06/2012 02:29 AM - Eric Faulhaber

I have made minor cleanups to the format and content and I have checked the updated chapter into bazaar.

The chapter already does a very good job of describing the index selection process. However, we need to add a section describing the converted outcome of this process (i.e., the selected index is used to generate a snippet of HQL, which becomes the "order by" clause at runtime, etc.). We need some examples of converted output. The samples do not need to be exhaustive; i.e., only enough query types and index selection scenarios to give a good flavor of the converted output. This chapter can refer to the Queries chapter of the book for further examples of converted queries, including these HQL phrases.

#2 - 08/06/2012 12:22 PM - Eric Faulhaber

- Status changed from New to WIP

- Assignee changed from Stanislav Lomany to Adrian Lungu

A good source for the samples already exists in the testcases/uast directory.

The index_supplemental.zip file contains compiler cross reference listings for a number of test cases, which are based on the p2j_test schema (see testcases/uast/data/p2j_test.df). For each *.xrf file in that archive, a corresponding *.p program exists in the testcases/uast directory. These programs were the basis of determining the query sorting rules for P2J.

Again, the examples do not need to be exhaustive; we are just looking for a representative sampling to indicate the practical outcome of the index selection process in terms of converted code.

#3 - 08/08/2012 01:50 PM - Adrian Lungu

- % Done changed from 0 to 100

#4 - 08/08/2012 01:54 PM - Adrian Lungu

I added the new section and committed into bazaar.

The issues related to content are marked with a TODO tag in the document itself.

#5 - 08/08/2012 11:23 PM - Eric Faulhaber

- % Done changed from 100 to 90

Excellent work so far. I've made some edits to the chapter, particularly in some of the example details, and I've checked it in again. However, some minor cleanup remains (see below).

First, some minor formatting/style issues...

Please capitalize all important words (i.e., all but articles like "a", "an", "the", etc.) in section headings. For instance, use:

```
Conversion Examples
```

instead of:

```
Conversion examples
```

Please use the "Teletype" style (from the "Character Styles" page of the "Styles and Formatting" dialog in LibreOffice) for inline references to language keywords or code. If in a table, use "Teletype Small".

Please use 2 spaces to separate sentences.

Always end a chapter with a manual page break. This is necessary when combining the chapters into a book.

In terms of content...

In the first example, the first line is:

```
[def-temp-table.i]
```

Is this a reference to the name of this include file?

The examples below that include a file as follows:

```
{person-temp-table.i}
```

...but this include file is never defined anywhere. I suspect it was meant to refer to the first snippet of code (with that [def-temp-table.i] line). If so, please indicate the name of the include file outside of the example. Something like:
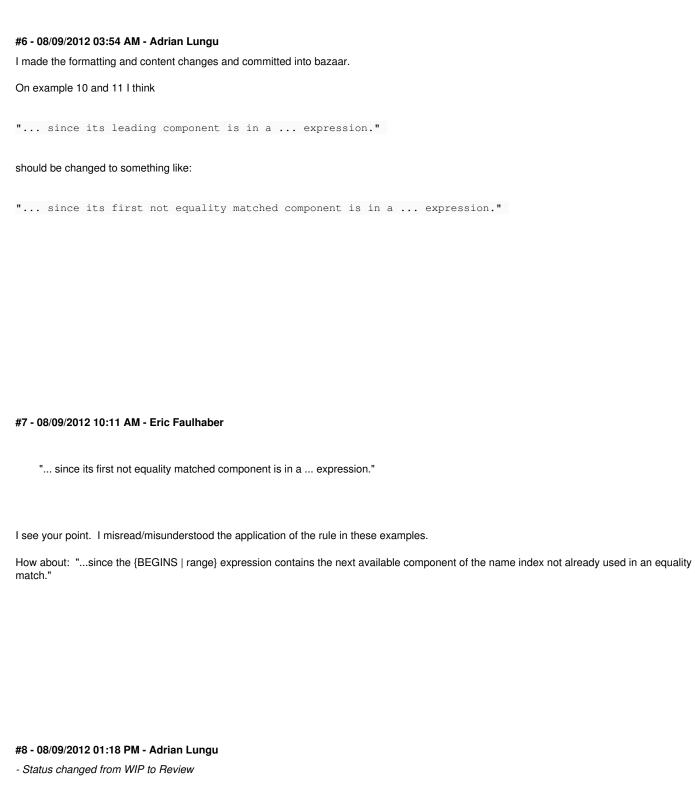
The following examples will use the ***person*** temporary table, as defined in **person-temp-table.i**:

```
/* person-temp-table.i */
define temp-table person
       field site-id as integer
       field emp-num as integer
       field first-name as character
       field last-name as character
       ...
```

Please remove example 16. It is confusing. 17 is enough to illustrate rule 8 (alphabetically first index), and it does so pretty cleanly. The remaining examples will need to be re-numbered, of course.

I agree with your inline comment that the last example can be removed.

#6 - 08/09/2012 03:54 AM - Adrian Lungu

I made the formatting and content changes and committed into bazaar.

On example 10 and 11 I think

`"... since its leading component is in a ... expression."`

should be changed to something like:

`"... since its first not equality matched component is in a ... expression."`

#7 - 08/09/2012 10:11 AM - Eric Faulhaber

> "... since its first not equality matched component is in a ... expression."

I see your point. I misread/misunderstood the application of the rule in these examples.

How about: "...since the {BEGINS | range} expression contains the next available component of the name index not already used in an equality match."

#8 - 08/09/2012 01:18 PM - Adrian Lungu

- Status changed from WIP to Review

I fixed example 10 and example 11 text and committed into bazaar.

`"... since its first not equality matched component is in a ... expression."`