Query Writing for FA Users

Session #26473
March 25, 2009
Anaheim, California

Your Presenter

- Brian Robson – University of Michigan
- Senior Business Systems Analyst for MAIS – Michigan Administrative Information Services
- 20 Years with the University of Michigan
- 12 Years as a Business Systems Analyst

University of Michigan Fast Facts

- Established 1817 - first public university in the Northwest Territories
- 3 campuses: Ann Arbor, Dearborn, Flint
- 19 schools and colleges, 24 libraries, 11 museums, 100+ centers and institutes
- 56,000+ students
- 58,000+ faculty and staff
- 524 major buildings = 35 million gross sq. ft.
- Health System (est. 1.6 M patients annually)

• Financial Picture
  - $4.5B annual budget
  - $823M research funding
  - $7.1B endowment

About MAIS

- The University of Michigan Administrative Information Services (MAIS) is responsible for the campus administrative computer systems' infrastructure and security
- MAIS services:
  - University Health System
  - 3 regional campuses (Ann Arbor, Flint, and Dearborn)
  - All research and educational units (35 centers and 18 institutes)

Administrative Applications

<table>
<thead>
<tr>
<th>PeopleSoft/Oracle</th>
<th>HE (HRMS/Student Admin) v9.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financials/Supply Chain v8.8</td>
<td>Campus Community</td>
</tr>
<tr>
<td>Accounts Payable</td>
<td>HR</td>
</tr>
<tr>
<td>Accounts Receivable</td>
<td>Benefits Administration</td>
</tr>
<tr>
<td>Asset Management</td>
<td>Time &amp; Labor</td>
</tr>
<tr>
<td>Billing</td>
<td>Payroll</td>
</tr>
<tr>
<td>eProcurement</td>
<td>Recruiting/Admissions</td>
</tr>
<tr>
<td>eSettlement</td>
<td>Student Records/Curriculum</td>
</tr>
<tr>
<td>General Ledger</td>
<td>Academic Advising</td>
</tr>
<tr>
<td>Inventory</td>
<td>Financial Aid</td>
</tr>
<tr>
<td>Purchasing</td>
<td>Student Financials</td>
</tr>
<tr>
<td>Space Management (homegrown)</td>
<td></td>
</tr>
</tbody>
</table>

Overview

- Present some basic and advanced query writing techniques, using FA tables.
- Goal: to get you comfortable and familiar with many aspects of this reporting tool.
- Examples in HE Version 9
- Examples will focus more on the tool than on the tables.
- Recommend attending: 26533-Financial Aid Data Structures
Selecting tables & fields

PSQuery is a tool that helps you create and run Structured Query Language (SQL) select statements.

```
SELECT A.EMPLID, A.ITEM_TYPE
FROM PS_STDNT_AWARDS A
WHERE A.AID_YEAR = '2009'
```
Selecting tables & fields

The Records Tab
- Tables must be included on your query security tree to be available.
- You must have a role/permission list that allows access to the query security tree.

The Query Tab

Select the key fields and other fields

The Fields Tab

Selecting tables & fields

Selecting tables & fields

Adding Criteria
- Add criteria to narrow your results

Adding Criteria
- Field AID_YEAR is equal to a Constant, ‘2009’
Adding Criteria
- The Criteria tab

Reading the SQL Statement
- The View SQL tab

Running the Query
- Results

Adding a Prompt
- Edit button allows you to change your criteria
- Click the 'New Prompt' button to create a prompt
- Select the properties for the prompt
SQL Statement with a Prompt
- :1 signifies a variable, or in PSQuery, a prompt

Using the Prompt
- When running the query, the user is prompted for the variable

Using the Prompt
- If a Prompt Table was specified, a list of values will be displayed

Column and Row Ordering
- On the Fields tab, click the Reorder/Sort button

Column and Row Ordering
- Change column order and row order (sort)

Column and Row Ordering
- Column and Sort order is displayed on the Fields tab
Column and Row Ordering
- Sorted Results

Counting and Summing Functions
- Add aggregate functions to fields by clicking the Edit button on the Fields tab

Counting and Summing Functions
- Select and aggregate

Counting and Summing Functions
- SQL statement with aggregate functions
  - PS adds a GROUP BY clause

Counting and Summing Functions
- Counting and Summing Results

‘Having’ Criteria
- To place criteria on an aggregated field, you must use ‘Having’ criteria
  - Scenario: You want to see which careers disbursed less than $15M this year.
  - Use ‘Having’ criteria on SUM(A.DISBURSED_AMOUNT)
‘Having’ Criteria
- On the Having Tab, click the Add Having Criteria button

‘Having’ Criteria
- Click the Choose Record and Field prompt

‘Having’ Criteria
- Aggregated fields will be displayed
- Select a field for having criteria

‘Having’ Criteria
- Define the criteria

‘Having’ Criteria
- Having Tab

‘Having’ Criteria
- Having SQL
Having Criteria

Results

Scenario: You want to find out which individual students received disbursements.

Distinct Select

Distinct select will return the unique rows for the fields that are selected.

Click the Properties link to make your query distinct

Distinct Select

Select the Distinct checkbox

Distinct Select

SQL

Distinct Results
Expressions
- Use an expression when the values you need are not readily available.
- Expressions can be added to either the select statement or the criteria.

Scenario: You want to know the difference between the total accepted amount and the total disbursed amount for a certain item type.
- The difference can be expressed as:
  - `SUM (A.ACCEPT_AMOUNT - A.DISBURSED_AMOUNT)`
- Use this expression in the select statement.

Expressions
- Click the Add Expression button

Expressions
- Enter the Expression Text

Expressions
- Click the Use as Field link

Expressions
- The expression becomes a field on the Fields tab
Expressions

- SQL Statement

![SQL Query]

Expressions

- Expressions can be added to the criteria as well:

![Criteria Properties]

Joining Tables

- Scenario: You want to find out what aid was disbursed to entering undergraduate students.

  - Join STDNT_FA_TERM to STDNT_AWARDS

  - STDNT_FA_TERM Criteria:
    - ACAD_LEVEL_BOT = 11
    - STRM = 1710

Joining Tables

- On the Records tab, find the table and click the Join Record link.

![Find an Existing Record]

Joining Tables

- Select the table to which you want to join

![Select Table]

Expressions

- Results
Joining Tables
- PS looks for Auto Join Criteria

Joining Tables
- The table is added to the query

Joining Tables
- Add Criteria from STDNT_FA_TERM

Joining Tables
- Results

Grouping Criteria
- Scenario: Now you want to find out what aid was disbursed to entering undergraduate students and sophomores with a certain minimum grade point average, but the grade point average is different for the two groups:
  Disbursed
  and
  (Freshman with GPA > 3.5
  or
  Sophomore with GPA > 3.8)

Grouping Criteria
- Add new criteria and click Group Criteria button
### Grouping Criteria
- Add Parenthesis to group criteria
- Will this grouping achieve the desired result? Hint: (no)

### Effective Date Criteria
- A closer look at this criteria
- STDNT_FA_TERM is an effective dated table
- PSQuery automatically created effect date criteria when STDNT_FA_TERM was chosen

### Effective Date Criteria
- PSQuery selects the current effective date/seq for you, and builds the necessary SQL.
- The SQL is two sub-queries written automatically.
Effective Date Criteria

- Effective date on Criteria Tab

- SQL for effective date

```
AND ED_C hội / ED / ED (ED) FROM PS_R router / ED (ED) WHERE B ED / ED / ED AND B STR / B / ED AND B ED / ED AND B / ED (ED) AND B / ED / ED = SYSDATE;
```

- Effective Sequence Options

Sub-queries

- A sub-query is an SQL SELECT statement that's nested inside of another SQL statement.

- Use sub-queries:
  - when you want to return a subset of the rows generated in your main query
  - when you have similar, but conflicting criteria on the same table/field (use an "exists" sub-query)
Sub-queries

- A sub-query is an SQL SELECT statement that’s nested inside of another SQL statement.

```
(SELECT MAX(B_ED_EFFECT) FROM B_ED_EMPLOY
AND B_INSTITUTION = B_ED_INSTITUTION
AND B_TERM = B_ED_STRM
AND B_ED_EFFECT = SYSDATE)
AND B.EFFECT =
(SELECT MAX(B.ED.EFFECT) FROM B_ED_EMPLOY
AND B_INSTITUTION = B_ED_INSTITUTION
AND B_TERM = B_ED_STRM
AND B.EFFECT = B.ED.EFFECT)
```

Sub-queries

- Also use sub-queries when you have similar, but conflicting criteria on the same table/field.

Scenario: You want to see students who have completed checklist ‘FLFYLN’, but have not completed checklist ‘F10REV’.

Sub-queries

- What’s wrong with this criteria from the PERSON_CHECKLST table?

```
The criteria for CHECKLIST_CD conflicts.
```

Sub-queries

- Create a EXISTS sub-query to find rows meeting certain criteria when additional, conflicting criteria exists.

Sub-queries

- Add the first set of criteria, then select ‘Add Criteria’ to build the EXISTS sub-query

```
Select condition type ‘exists’ and click Define/Edit Subquery
```

Sub-queries

- Select condition type ‘exists’ and click Define/Edit Subquery
Sub-queries

- You’ll be prompted to select a record for the sub-query

- You MUST add join criteria from the new table to a table in the main selection of the query

- Join field(s) to field(s) between the main query tables and the sub-query table

- The prompt lets you select from any of the tables currently in the query.

- Joining fields: COMMON_ID in the sub-query to COMMON_ID in the main query

- Add the remaining criteria for the sub-query
Sub-queries
- SQL for Exists

Sub-queries
- You can get back to the main query using the Subquery/Union Navigation link

Sub-queries
- Results contain data for students who have completed checklist ‘FLFYLN’, but have not completed checklist ‘F10REV’.

Outer Joins
- Scenario: You want to see students who have a disbursement and also have an active service indicator. You want the emplid to be returned even if there is no service indicator.
- Use an outer join to return rows on multiple tables even if there is not a match between the joined tables.
- Psquery can add Left Outer Join automatically
Outer Joins

- Select “Join to get additional fields only”

- Left Outer Join SQL

Unions

- Use a UNION to combine similar data from different tables into single columns in the results.

- A UNION combines the results of two SQL queries into a single table of all matching rows. Duplicate rows are eliminated.

Unions

- Scenario: You want to see the record of award disbursement and account posting by date/time.

- The history of the transactions resides on two tables with similar fields.

- A UNION can return one list, sorted by dttm
Unions

- STDNT_AWARD_ACTV has the FA disbursement activity: emplid, dtmm, item_type, award_disb_action and amount.

- ITEM_LINE_SF has the SF posting activity: common_id, dtmm, item_type, line_action and amount.

Unions

- Create the first query, then click “New Union.”

Unions

- Create the second query, using corresponding fields.

Union SQL:

Unions

- UNION results

Questions/Discussion

- Notes from the 3/25/09 discussion:
  - Outer Joins:
    - Only the last table being joined can be automatically outer-joined.
    - Criteria added after the auto-join will not be outer joined; you must add expressions with (+) to avoid a standard join.
  - Sub-queries:
    - PSQuery will not prompt you with auto-join criteria when you select a table for your sub-query, but, you MUST join your sub-query table to one of your main query tables to avoid a run-away query.

(Notes continue on next slide)
Questions/Discussion

Notes from the 3/25/09 discussion (cont’d.):
- A query can be copied to another user on the main Query Manager page.
- Note: A copied query cannot be deleted by the user to whom the query was copied.

Contact

- Brian Robson
- Senior Business Systems Analyst
- Michigan Administrative Information Services
- University of Michigan
- E-mail: brobson@umich.edu

Prizes!!

Complete Evaluation to Win!

8 iPod Shuffles
(one with 4GB given away at Wed a.m. Board Meeting)

Daily Drawing for the winners

Just Complete the Online Evaluation

http://www.heug.org/p/su/rd/sid=283

This presentation and all Alliance 2009 presentations are available for download from the Conference Site

Presentations from previous meetings are also available