

Calculations in IRDB/Discoverer

IR Council

October 28, 2011

Review

- Use a Table to make a data set to export and work with outside the IRDB
- Use a cross tab to create summary results
- Business Areas
- Folder=Table
- Item=Field
- Joins

Review

- Row item(s)
- Column item(s)
- Page item(s)
- Data point(s)
- Condition(s)
- Calculation(s)...

Why make calculations?

- Convenience: Once created in Discoverer, calculations can be used for multiple queries in the workbook (create it once), and copied from one workbook to another
- Apply standard methodology across multiple workbooks/queries
- Enables flexible presentation of query results
 - Compare cross tab using categorical field and calculating in Excel with creating calculation in discover
- Show multiple data points in one query, even based on different populations

Types of Calculations

- Categorical from single item – re-categorizing
- Categorical from multiple items – more complex categories
- Data point

How to determine what type to create

Table (detailed data)

- Flag
- Dummy variables

Cross Tab (summary data)

- Counting variable (sum on flag)
- Percentage
- Average

Preparation for Calculations

1. Define your population – usually from (at least) the main table in the business area – by setting implicit and/or explicit conditions
 - Select term(s)
 - Select college
 - Select other values to select relevant group (e.g., History Facts.New Student Code='1' to identify first-time freshmen)
2. Check the N's to make sure you've defined your group appropriately
3. Select at least one item from each folder from which your calculation will be created (this will insure that the folder/item is available for selection when building your calculation)
4. Think of a descriptive name for your calculation that will distinguish it from other calculations you plan to make (e.g., 6-yr system grad rate; 4-yr institution grad rate)
5. You can reference calculations in other calculations within the same Discoverer workbook

Writing Calculations: Simple Categorical

- **CASE WHEN** *folder.item* [*CONDITION*] **THEN** [*RESULT1*] **WHEN** *folder.item* [*CONDITION*] **THEN** [*RESULT2*] ... *folder.item* [*CONDITION*] **THEN** [*RESULTn*] **ELSE** [*ALTERNATE RESULT*] **END**

Writing Calculations: Complex Categorical

- **CASE WHEN** *folder1.item1* [CONDITION] **AND** *folder1.item2* [CONDITION] **AND** *folder2.item1* [CONDITION] **THEN** [RESULT1] **WHEN** *folder1.item1* **AND** *folder1.item2* [CONDITION] **THEN** [RESULT2] ... *folderN.itemN* [CONDITION] **THEN** [RESULTn] **ELSE** [ALTERNATE RESULT] **END**

Writing Calculations: Data Points

- **Flag (detail):**
 - **CASE WHEN** *folder.item* [*CONDITION*] **THEN 1 ELSE NULL END**
- **Counting calculations (summary):**
 - **SUM(CASE WHEN** *folder.item* [*CONDITION*] **THEN 1 ELSE NULL END)**
 - **SUM(CASE WHEN** *folder.item* [*CONDITION*] **THEN** *History Facts.Headcount* **ELSE NULL END)**

Writing Calculations: Data Points

- **Averages (summary):**
 - **SUM(CASE WHEN History Facts.Full Part Type Code='1' THEN History Facts.Semester Credits Earned Total Perf ELSE NULL END)/SUM(CASE WHEN History Facts.Full Part Type Code='1' THEN History Facts.Headcount ELSE NULL END)**

Writing Calculations: Data Points

- **Percentage (summary):**

EXAMPLE: % Full Time

- $100 * \text{SUM}(\text{CASE WHEN History Facts.Full Part Type Code}='1' \text{ THEN History Facts.Headcount ELSE NULL END}) / \text{SUM}(\text{History Facts.Headcount})$

- **Percentage based on known (summary):**

EXAMPLE: % Not Proficient

- $100 * \text{SUM}(\text{CASE WHEN History SKAT Initial Facts.All Skill Pass Level Code IN ('0','1','2') THEN History Facts.Headcount ELSE NULL END}) / \text{SUM}(\text{CASE WHEN History SKAT Initial Facts.All Skill Pass Level Code IN ('0','1','2','3') THEN History Facts.Headcount ELSE NULL END})$

Cautions: 0 vs NULL

- When creating aggregate data points, consider setting ELSE = NULL rather than ELSE=0 in case you use this calculation as the denominator of a percentage or average – if the sum evaluates to 0, the calculation will not evaluate. If the denominator evaluates to NULL, the calculation will evaluate to NULL