Introduction to Reporting using Banner Tables
(Student Module)
<table>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Information .................................................. 4</td>
</tr>
<tr>
<td>Chapter 1: Setting the Reporting Stage ............................ 5</td>
</tr>
<tr>
<td>Chapter 2: Getting Started ........................................... 7</td>
</tr>
<tr>
<td>Defining the Business Question ...................................... 7</td>
</tr>
<tr>
<td>Create a Query to Answer the Business Question .................. 8</td>
</tr>
<tr>
<td>Chapter 3: Identifying Data within Banner ....................... 13</td>
</tr>
<tr>
<td>Dynamic Help Query .................................................... 13</td>
</tr>
<tr>
<td>Data Schema and Entity Relationship Diagrams ................... 15</td>
</tr>
<tr>
<td>Catalog Pane ........................................................... 19</td>
</tr>
<tr>
<td>Chapter 4: Creating the Data Model .................................. 23</td>
</tr>
<tr>
<td>Chapter 5: Populating the Request Line ............................. 28</td>
</tr>
<tr>
<td>Build a Request line .................................................... 28</td>
</tr>
<tr>
<td>Processing a Hyperion Query .......................................... 30</td>
</tr>
<tr>
<td>Chapter 6: Limiting the Query ....................................... 34</td>
</tr>
<tr>
<td>Applying a Limit ......................................................... 34</td>
</tr>
<tr>
<td>Adding Additional Limits ............................................... 37</td>
</tr>
<tr>
<td>Chapter 7: Aggregating the Query ................................... 40</td>
</tr>
<tr>
<td>Define Calculations ..................................................... 40</td>
</tr>
<tr>
<td>Adding an Aggregate Calculation to the Query .................... 41</td>
</tr>
<tr>
<td>Chapter 8: Creating Join Relationships ............................ 45</td>
</tr>
<tr>
<td>Building a Join .......................................................... 45</td>
</tr>
<tr>
<td>Auto-Joins ............................................................... 47</td>
</tr>
<tr>
<td>Join Rules and Validation .............................................. 49</td>
</tr>
<tr>
<td>Chapter 9: Sorting the Results ....................................... 51</td>
</tr>
</tbody>
</table>
Table of Contents (continued)

Chapter 10: Formatting the Results ........................................ 53
   Moving Columns ......................................................... 53
   Rename Result Columns ............................................... 54
   Formatting Numbers .................................................... 56
   Resizing Result Columns .............................................. 57
   Hiding the Sections/Catalog Pane ..................................... 58
   Suppressing Duplicates ................................................ 58
   Adding Grand Totals to Columns ..................................... 59
   Adding Subtotals ....................................................... 61
   Aligning Data .......................................................... 62

Chapter 11: Managing Hyperion Documents ...................... 63
   Exporting the Result Set ............................................... 63
   Inactivating Save Query Results from the Document .......... 64

Appendix A: Final Exercise ............................................. 65

Appendix B: Tables ....................................................... 68
General Information

Purpose

The course introduces all attendees to the Hyperion Intelligence 8 reporting tool and demonstrates general tool functions, such as creating basic queries, building formatted reports and exporting information to Excel. The course enables all attendees to perform hands-on exercises derived from relevant scenarios and also presents common reporting mistakes that attendees should strive to avoid when working with the data.

Audience

This course is designed for all University personnel that will use the Banner database and Hyperion reporting tool. Participants should have a functional knowledge of the data within their module of Banner (i.e. Student) in order to effectively take part in the class.

Learning Outcomes

On completion of this course, you will be able to:

- Understand the data aspects stored within Banner.
- Launch Hyperion and build a simple query in order to answer a business question
- Format query results to meet specific presentation requirements
- Save, export and share Hyperion report documents
Chapter 1: Setting the Reporting Stage

All query and reporting activities performed using Banner originate out of business questions, job requirements or requests for information. In some cases, these business questions are repeatable in nature and will be satisfied by a consistent report that refreshes on a regular interval, such as once per day or once per month. In other instances, these questions are variable and change from day to day, semester to semester, and academic year to academic year.

Databases

One of the first aspects to understand is what is behind Banner; and that is the database structure. The web client allows you to retrieve data from relational databases and multidimensional databases. Once you connect to a database, you can build a query to retrieve data to your desktop, and then work with the data in report sections.

Banner uses a relational database structure.

Relational databases store information in tables. A table is subdivided by column fields that group related information. The column fields are children of the parent table. For example, the Customer table may have columns including Name, Address, and ID number. Tables appear as topics, and columns appear as topic items. Each table contains row records that describe information about a singular entity, object, or event.

Database tables are linked by joins. Joins allows row records in different tables to linked based on shared information in a column field. For example, a row record in the Customer table joins to a related record in the Orders table when the Customer ID value for the record is the same in each table. This allows the order record to be linked with the record of the customer who placed the order.

Banner Environments

At the University of Memphis, there are multiple Banner environments a user can access depending on the work to be done.

There are three primary environments/instances are:

- **ITRNG**
  - Used by functional and technical staff for internal UofM training. Contains institution-specific data to be used for institutional training and testing data and configuration. Cloned from an instance whose purpose is to refresh other instances.

- **RPTS**
  - Only accessed via Hyperion (Brio)
  - Cloned from PROD daily from the daily backups.
- TBANR
  - Used by functional and technical staff to validate data, rules and processes. Development, testing and validation of data load processes, BANNER functionality, local modifications, and interfaces are conducted within this instance. Cloned from production periodically.

Do these questions sound familiar?

The following examples demonstrate business questions and reporting requests supported by Banner tables. As previously mentioned, the true list is infinite based on the point in time and all combinations of data elements, calculations and conditions. These examples provide real world examples in how Banner tables can be utilized.

Enrollment

- Number of students enrolled by level and student type
- Courses and status of those courses for students identified as veterans
- Students registered for audit courses

Admissions

- Number of applicants for a specific term by level, admit type and curriculum priority
- Applicants with “MS-Desoto” county and residency type of “2”
- Applicants accepted for specific term with A89 deficiencies (SZAADEF – A89)

Financial Aid

- All new undergraduate freshmen who graduated from high school on or after May 01, 2007, who are Pell Grant eligible, are U.S. citizens and meet the rigorous high school curriculum standards
- Open ‘REVW’, ‘REVWD’, ‘REVWI’ on our document tracking form (i.e. RRAAREQ) sorted by date and counselor
- Financial aid applicants who are pursuing an Undergraduate degree for the current academic year, who have at least a 2.5 cumulative GPA, show financial need, and are majoring in math, science or engineering
Chapter 2: Getting Started

A query is a set of instructions that retrieves and manipulates data from the database in order to answer a specific business question. Queries consist of information elements to be retrieved, calculations to be performed, sort orders to be applied and restrictions that narrow the final result set.

Section I – Defining the Business Question

Every query corresponds to a business question, job requirement or request for information. Before building a query in the Hyperion, first analyze the business driver for the query to understand its component parts. The initial business question in its entirety may seem overwhelming, but when broken down into component parts, it becomes easier to translate into a query.

Assume the following business question:

Generate a document that counts the enrolled students by gender. This list should only include those that are registered at the U of M for Fall 2007 semester. It should also be sorted by gender.

Most business questions include one or many of the following component parts:

- **Elements to be retrieved** – the list of attributes and measures included in the envisioned final result
- **Restrictions that govern the final result set** – what filters should be applied in order to retrieve the right results
- **Sort orders to be applied** – how should the results be sorted to promote readability
- **Calculations to be performed** – what calculations should be performed on attributes and/or measures to enhance the result set
What are the component parts in the sample business question?

**ELEMENTS TO BE RETRIEVED**
- Identify elements that should be displayed in your final result

**GENDER**

**RESTRICTIONS THAT GOVERN THE FINAL RESULT SET**
- Identify any component part that limits the result qualifying one or many elements.

**THIS LIST SHOULD ONLY INCLUDE THOSE THAT ARE REGISTERED AT THE U OF M FOR THIS SEMESTER.**

**GENERATE A DOCUMENT THAT COUNTS THE ENROLLED STUDENTS BY GENDER. THIS LIST SHOULD ONLY INCLUDE THOSE THAT ARE REGISTERED AT THE U OF M FOR THIS SEMESTER. IT SHOULD ALSO BE SORTED BY GENDER.**

**SORT ORDERS TO BE APPLIED**
- Should the result be sorted to promote readability?

**SORTED BY GENDER**

**COUNT THE ENROLLED STUDENTS**

**RESTRICTIONS THAT GOVERN THE FINAL RESULT SET**
- Identify any component part that limits the result qualifying one or many elements.

**CALCULATIONS TO BE PERFORMED**
- Identify requirements for counting, summing, averaging, multiplying, dividing, subtracting, etc…

**Section 2 - Create a Query to Answer the Business Question**

After understanding the component parts that comprise the business question, the question should be translated into a query.

The steps to follow when creating a query in Hyperion are:

1. Select the Hyperion Intelligence “Blank Document”
2. Identify the tables required to satisfy the component parts within the business question
3. Understand how the tables relate to each other
4. Define a data model
5. Build a request line
6. Build a limit line
7. Build a sort line
8. Define calculations based on the business problem
9. Process the query
10. Format the result set

Hyperion contains a report catalog where existing report documents may be stored and accessed by users. The catalog contains category folders corresponding to each functional area within Banner under the Spectrum folder. This folder also contains a blank .bqy document titled Blank Doc for Spectrum - ITRNG - User Specific. This document lists all tables that may be accessed according to user login security and does not include any existing query or report definitions. This document should be used to create most new ad-hoc queries and reports.

Select the Hyperion “Blank Document”

1. Log in to Hyperion with UUID and password at: http://www.memphis.edu/dwlogin.

2. Navigate to the Spectrum folder and double click to expand.
3. Click **Blank Doc for Spectrum - ITRNG - User Specific** to open the .bqy document. The blank document will open and this will be used to construct a new query to answer the business question.

If nothing happens when clicking on the file you will need to install the Hyperion plug-in on your machine. Information for this can be found by logging into Hyperion (http://www.memphis.edu/dwlogin) and going to **Hyperion Intelligence plug-in Install and Uninstall Guide** within the **Browse** screen and clicking on the pdf.

A blank Hyperion document:
Saving and Accessing the Hyperion Document

Hyperion documents may be saved to a local folder or a shared network folder as .bqy files so that they may be re-used or refreshed at a later date.

4. In the toolbar, select File > Save As.

   The Save File window will appear.

   ![Save File Window]

5. Navigate to your local account’s desktop folder and name the document *Enrolled by Gender*. Click Save in order to save the current version of the .bqy file.

   You are now ready to go to your desktop to access the .bqy file.

   ![Desktop Folder]

   Any existing Hyperion document must be associated with an Internet Explorer page prior to opening. In order to launch an existing document from a local or shared file folder location:
6. On the desktop, left-click once on the saved .bqy file to select it.

7. Right-click once to launch the shortcut menu, select **Open With...**, then **Internet Explorer**. If **Internet Explorer** is not listed the **Recommended Program** window will appear (see below).

Select **Internet Explorer** under **Recommended Programs**:

Place a checkmark in the box **Always use the selected program to open this kind of file**.

Click **OK**. Hyperion will be launched.

8. Login to the Hyperion screen that was launched using your UUID and password. This opens the selected document within an Internet Explorer browser session.
Chapter 3: Identifying Data within Banner

Identifying data within the Banner tables can, at times, be a complex task. However, there are a few tools to assist you if you know where to look.

Section 1 – Dynamic Help Query

1. Within Banner (accessed through the Spectrum portal), you will need to be within a form. This can be any form (i.e. SPAIDEN, SOAHOLOD, etc...).

2. Using the Next Block icon in the tool bar, get your cursor into the lower portion of the screen and then select the field you would like more information on (i.e. Address tab > Street Line 1).

3. Once your cursor is within the appropriate field, click Help in the tool bar > Dynamic Help Query.

The GUAHELP box will appear.
4. Within the GUAHELP box, take note of the Block and the Field title listed. The Block title and first section of the Field title should be the name of the table this field appears in (A).

For example, if the Block is SPRADDR and the Field is SPRADDR_STREET_LINE1 this field should be located in the table SPRADDR. To be sure that it is in the table refer to the list of the Tables Common to all Banner Modules below.

While the Block and the Field can represent the table name and column name respectively, to show additional data click the Banner radio button, then Display. The Help Text box will display.

Help Text box:

```
FIELD DESCRIPTION
This field maintains the first line of the address associated with person.
USAGE NOTES
Address(Line One)
The Address(Line One) field is used to maintain the first line of the address associated with person/non-person.
NOTE: DO NOT put a "#" (pound sign) in the address. This will cause problems with commands in the form GUALETR.
```
Tables Common to All Banner Modules

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Description</th>
<th>Sample Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOREMAL</td>
<td>Email Address Table</td>
<td>Email Address</td>
</tr>
<tr>
<td>GURMAIL</td>
<td>Mail Repeating Table</td>
<td>Letter Codes, Print Dates</td>
</tr>
<tr>
<td>SPBPERS</td>
<td>Basic Person Base Table</td>
<td>Sex, Ethnicity, Birth Date, SSN, Deceased Indicator, Confidentiality</td>
</tr>
<tr>
<td>SPRADDR</td>
<td>Address Repeating Table</td>
<td>Street 1, 2, 3, City, State, Zip, Nation</td>
</tr>
<tr>
<td>SPRIDEN</td>
<td>Person Identification / Name Repeating Table</td>
<td>ID, Last Name, First Name, Middle</td>
</tr>
<tr>
<td>SPRTELE</td>
<td>Telephone Table</td>
<td>Area Code, Phone Number</td>
</tr>
</tbody>
</table>

*Additional tables can be found in Appendix B*

After you have confirmed the name of the table there are two ways of seeing the information:
- Data Schema/Entity Relationship Diagrams on UMdrive
- Show Remarks and Properties fields within the tables in Hyperion

Section 2 – Data Schema and Entity Relationship Diagrams (ERD)

Database schema is a collection of meta-data that describes the relations in a database. A schema can be simply described as the "layout" of a database or the blueprint that outlines the way data is organized into tables.

Entity Relationship Diagrams (ERDs) present the underlying database table relationships in Banner. An entity relationship diagram is a data modeling technique that creates a graphical representation of the entities (tables), and the relationships between the entities (tables), within an information system.

With the exception of finance and accounts receivable, both Data Schema and ERDs exist for the Banner tables. Finance and accounts receivable have only ERDs.

5. Go to UMDrive (https://umdrive.memphis.edu/xythoswfs/webui) and log in using your UUID and password.
6. Navigate to /g-banner/data_quality/ERD/7.3 >
the file ERD_Index.html; open the file.

7. On the website that displays, select Student, then the Sungard SCT ERD website is displayed.
[html] will display the Data Schema such as name, width, key(s), and a description of each column within the table. [pdf] displays the ERD of how the tables graphically relate to each other.

Click [html] next to General Person. The CAST HTML report will display.

8. Scroll down and under the List of Objects select the table SPBPERS.

Below is information about ERDs in HTML format:

**ERDs - HTML format**

- **Table Columns information**
  - Column information for each table is displayed including the primary key designation.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Label</th>
<th>Column Type</th>
<th>Allow NULL</th>
<th>PK FK</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPBPERS_PDM</td>
<td>SPBPERS_PDM</td>
<td>NUMBER(10)</td>
<td>No</td>
<td>*</td>
</tr>
<tr>
<td>SPBPERS_PSN</td>
<td>SPBPERS_PSN</td>
<td>VARCHAR(20)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>SPBPERS_BIRTH_DATE</td>
<td>SPBPERS_BIRTH_DATE</td>
<td>DATE</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>SPBPERS_LOCY_CODE</td>
<td>SPBPERS_LOCY_CODE</td>
<td>VARCHAR(20)</td>
<td>Yes</td>
<td>*</td>
</tr>
<tr>
<td>SPBPERS_ETHN_CODE</td>
<td>SPBPERS_ETHN_CODE</td>
<td>VARCHAR(20)</td>
<td>Yes</td>
<td>*</td>
</tr>
<tr>
<td>SPBPERS_MRTL_CODE</td>
<td>SPBPERS_MRTL_CODE</td>
<td>VARCHAR(20)</td>
<td>Yes</td>
<td>*</td>
</tr>
<tr>
<td>SPBPERS_RLEG_CODE</td>
<td>SPBPERS_RLEG_CODE</td>
<td>VARCHAR(20)</td>
<td>Yes</td>
<td>*</td>
</tr>
</tbody>
</table>

- **Foreign Key information**
  - Section and tables associated with columns in the current table are displayed.

<table>
<thead>
<tr>
<th>Column name</th>
<th>is Foreign Key For Column</th>
<th>on Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPBPERS_PDM</td>
<td>GOREND_TLP_PDM</td>
<td>GOREND</td>
</tr>
<tr>
<td>SPBPERS_PSN</td>
<td>GORESVS_PSN</td>
<td>GORESVS</td>
</tr>
<tr>
<td>SPBPERS_LOCY_CODE</td>
<td>STUYLOCY_CODE</td>
<td>STUYLOCY</td>
</tr>
<tr>
<td>SPBPERS_ETHN_CODE</td>
<td>STUYETHN_CODE</td>
<td>STUYETHN</td>
</tr>
<tr>
<td>SPBPERS_MRTL_CODE</td>
<td>STUYMRTL_CODE</td>
<td>STUYMRTL</td>
</tr>
<tr>
<td>SPBPERS_RLEG_CODE</td>
<td>STUYRLEG_CODE</td>
<td>STUYRLEG</td>
</tr>
</tbody>
</table>

- **Object Dates, Source Code Comments**
  - The description of the table is displayed.

**Object Dates**

<table>
<thead>
<tr>
<th>Creation Date</th>
<th>Analysis Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/13/2005 01:46:40.880</td>
<td>06/06/2006 05:26:29.000</td>
</tr>
</tbody>
</table>

**Source Code Comments**

Basic Person Base Table
9. Go back to the Sungard SCT ERD web page, and select the [pdf] next to General Person (see step 5).

**Column Comments, Source Code Comments**

- The description of the columns is displayed.

**Columns Comments for SPBPERS_PIDM (#1)**

**Source Code Comments**

Internal Identification Number of Person

---

**ERDs - PDF format**

In general, the "join" links are relationships that come from relationships that are found from the client code (Oracle Forms), not physically defined in the database. The links are imported into the database to help understand some of the "implied" links between tables or views in the database.

A red arrow means an actual link and its direction indicates a "calling" relationship. At one end, the object is calling, at the other end, the object is called.
Section 3 - Catalog Pane

While the process of using the Hyperion category pane will briefly be discussed here, further explanation will be given in chapter three.

10. Within Hyperion open a blank Hyperion document. Once in the document, click in the Catalog Pane and open the Tables folder. The list of tables will display within the Catalog Pane.

**NOTE:** When clicking in the Catalog Pane for the first time a dialog box asking for your UUID and password will appear. Enter this and continue on with opening the Tables folder. Also, after periods of inactivity you will be asked to enter your UUID and password again.
11. Within the Catalog Pane’s list of tables, double click on the table you found from the Dynamic Help Query and/or the Commonly Used Table list from earlier; a box will appear with a list of the table’s fields.
12. Within the table’s list of fields, find the particular field you would like more information. Right click on the field name.

**NOTE:** Show Remarks and Properties can also be selected for the table name itself, however, little information will be given for it.

After you right click:
- select **Show Remarks** (A) to view what the field contains
- select **Properties** (B) to view the length of the field
In Class Exercise – Chapter 3

Use the Dynamic Help Link in Banner, the ERD on UMdrive, and/or the “Show Remarks” Option in Hyperion to answer the following questions.

1. In what table and field does a person’s birth date reside?
   - Table Name: _______________
   - Field Name: _______________

2. What’s the type and length for the birth date field?
   - Type: _______________
   - Length: _______________

3. What three values are allowed for the gender field within Banner?
   ___________________________________________
   ___________________________________________
   ___________________________________________

Answers:
1. Within Banner INB, go to SPAIDEN and populate the form with your data. Click on the Biographical tab. Click within the Birth Date field. Select Help -> Dynamic Help Query.
   - Table Name – SPBPERS
   - Field Name – SPBPERS_BIRTHDATE

2. Follow the instructions for each:
   a. Go to the UMdrive ERD folder, click on Student, General Person [html], SPBPERS. Scroll to SPBPERS_BIRTHDATE. Type - Date
   b. Insert New Query. Open Catalog pane to display tables, Drag SPBPERS to the Data Model. Click on Spbpers Birth Date and right-click. Select Properties. Length – 7

3. Within the New Query created in 2b, click on SPBPERS_SEX and right-click. Select Show Remarks. Valid Values are M - Male, F - Female, N - Unknown.
Chapter 4: Creating the Data Model

Define a Data Model

A data model is a graphical representation of the actual database tables and table relationships required to resolve a query. The Hyperion interface contains a Catalog Pane that includes a list of the tables available to the current user’s UUID.

1. Expand the Tables folder within the Catalog Pane by clicking on the + symbol next to the Tables folder. This opens the available table list within the Catalog Pane.

![Catalog Pane](https://example.com/catalog-pane.png)

In order to build a data model, tables must first be added to the Content Pane. The content pane is a working area within the document where all in-scope tables and the join relationships between tables are defined. Tables may be added to the content pane using one of the following techniques:

- **Highlight** the table in the Catalog Pane and drag it into the Content Pane
- **Double click** the table, and the table will be added to the Content pane
- **Right-click** the table and select Add Selected Items
2. The sample business question requires the student must be registered for the fall semester. A student’s registration can be found within SATURN.SFRSTCR, the Student Course Registration Repeating table. An additional table will be needed for the gender, but for the moment, let’s concentrate on the student table.

Add the SFRSTCR table to the Content Pane.

The above screen shot is an example of a simple data model. The table appears in Structure View within Hyperion, which enables viewing the table name and the columns included in the table.

What is a Topic?

Hyperion refers to tables as Topics and columns as Items once they are included in a data model. This training manual will continue to refer to tables as tables and columns as columns; however, the Hyperion documentation and menus often use the terms “topic” and “item.”
Sometimes, viewing the actual data included in the table is helpful when building a new query. By changing the table view to **Detail View**, a sampling of rows are retrieved and displayed in the content pane.

3. Click on the **SFRSTCR** table in the content pane, then right-click and select **Detail View**. The Detail View window for the table opens (see below).

![Detail View](image)

**Detail View:**

<table>
<thead>
<tr>
<th>SFRSTCR Term Code</th>
<th>SFRSTCR PID</th>
<th>SFRSTCR CIN</th>
<th>SFRSTCR Class Sort Key</th>
<th>SFRSTCR Reg Seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007080</td>
<td>157059</td>
<td>B28544</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2007080</td>
<td>96657</td>
<td>B2884</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2007080</td>
<td>100301</td>
<td>B28588</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2007080</td>
<td>100312</td>
<td>B28512</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2007080</td>
<td>121659</td>
<td>B28420</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>2007800</td>
<td>222212</td>
<td>B28581</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>2007800</td>
<td>182813</td>
<td>B28325</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2007800</td>
<td>222212</td>
<td>B28547</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2007800</td>
<td>222212</td>
<td>B2551</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

The **Detail View** option enables a view of the first 10 records included in a table. This is useful when determining the data type, format or content for columns in the table.
Removing Tables from the Content Pane

Tables may be removed from the content pane and data model by doing the following:

- Select the table header in the content pane and strike Delete on the keyboard.
- Select the table header in the content pane, right-click, and select Remove.
In Class Exercise – Chapter 4

1. Include the SGBSTDN table in the Content Pane and use the table Detail View to browse the first 10 records in SGBSTDN, the Base Student table. List a sampling of college code values (Hint: sgbstdn coll code 1).

2. What does sgbstdn coll code value: EN represent in the SGBSTDN? (Hint: Include the STVCOLL table in the Content Pane and use Detail View to view the data.)

3. Remove tables SGBSTDN and STVCOLL from the Content Pane.

Answers:
1. AC, AS, ED, EN
2. Herff College of Engineering
3. Select the table headers for each table in the content pane and press Delete on the keyboard.
Chapter 5: Populating the Request Line

Section 1 - Build a Request Line

The Hyperion Request line is a graphical representation of the final query result set. All columns that are included on the Hyperion Request line will be included in the query syntax that is sent to the database and will be displayed as columns in the final result set.

Adding Columns to the Request line

Columns may be added to the request line by doing the following:

- Click on the column in the data model, right-click and select Add Selected Items.
- Click and drag the column from the table to the request line.

1. Our sample business question requires the result set to display the number of registered students for the fall semester by gender. Using one of the methods above, add the following fields:
   - SFRSTCR_TERM_CODE
   - SFRSTCR_PIDM
   - SFRSTCR_CRN
   - SFRSTCR_PTRM_CODE
   - SFRSTCR_RSTS_CODE
   - SFRSTCR_BILL_HR
   - SFRSTCR_ADD_DATE
   - SFRSTCR_ACTIVITY_DATE

The column order on the request line will not necessarily dictate the order in the result set. Changing the column order in the query results will be introduced in a later section.
Estimating Query Size

To estimate the number of rows that will be retrieved by a Hyperion query:

2. In the toolbar, select Query > Estimate Query Size.

   The Query Count box will appear (see below).

   ![Query Count Box]

   This asks the tool to estimate the number of rows that will be retrieved by the defined query. Hyperion displays the estimated result in a dialog window.

   **Query Count:**

   ![Query Count Dialog]

   3. Click OK to close the Query Count box after examining the estimated result set size.

      **NOTE:** Actual rows returned may vary.

Removing Columns from the Request Line

Columns may be removed from the request line by doing the following:

- **Individual column removal:** Click to select the column on the request line, right-click and select Remove.
- **All column removal:** Highlight the word Request on the request line, right-click and select Remove.
Section 2 - Processing a Hyperion Query

After creating the query, it must be sent to the database for execution. This is accomplished by processing the query in Hyperion. When processing, the tool converts the query to database syntax, sends that syntax to the target database and initiates query execution. The database executes the query and returns the result set to Hyperion.

### Query Governors

Before processing a query, it is good practice to estimate the number of rows a query will return or limit the initial result set to a sampling of the complete results. This presents an idea of the query magnitude, allows spot-checking of the complete results and avoids “runaway queries” that take long periods of time to execute.

View Query Properties

The **Query Properties** dialog box enables access to additional query governors. To retrieve a sampling of the complete result set for a defined query:

1. Open the **Query Properties** window by double-clicking on the word **Request** on the Request line or by selecting **Query** in the toolbar > **Query Options**. The Query Properties box will appear:

   ![Query Properties Window](image)

   - **Return Unique Rows** - removes duplicate rows from the retrieved result set.
   - **Return First X Rows** - limits the result set to the first x rows for the defined query. This enables retrieval of a data sample in order validate the query definition.
   - **Time Limit X Minutes** - enforces a configurable maximum time limit for query process time. If the query hits the maximum time limit, Hyperion will terminate the query and cease interaction with the database. The decimal precision in this number represents seconds.

The **Query Properties** window enables the following query governors:
Process the Query

After validating the query definition using a governor technique, the query is ready to be processed. Upon clicking the **Process** button on the application toolbar, Hyperion will derive the appropriate SQL syntax based on the query definition and send that syntax to the database. This action will eventually retrieve the query result set in the Hyperion **Results** section.

5. Within the toolbar, click the **Process** button above the request line.

Processing the query yields the following result set in the results section:
The bottom right hand corner of the results section captures two important pieces of information: the result set **row count** and **last process date/time**.

**NOTE:** Actual number of rows retrieved may vary.

**Understanding the Result Set with Banner Tables**

There are several items of interest to note about this result set that are applicable to most Banner base tables:

- Much of the data is at a transaction level, meaning there is a great deal of detail. In this case, there are multiple rows per the primary internal ID or PIDM because students can, in most cases, register for more than one courses or PIDM. Many tables will also have an activity date and/or add date and/or effective date for the entry of that row of data.

- Descriptions often do not accompany the codes within this table (ie RW for registration status in the SFRSTCR_RSTS_CODE ). In most cases a validation table must be joined to in order to retrieve the description. Validation tables are generally recognized by the V within the third position of the table name.

- Finally, as a result of being a relational database, data such as gender is not contained within the table as it is not a necessity or requirement for the process of storing registration data. This item is instead contained within a separate table which can be referenced through a join on the PIDM.

**Careful Validation is Key**

All query results must be carefully verified and validated against result expectations. Every query and the formatted reports built using the query results are only as good as the underlying data. Spot-checking against Spectrum (aka Banner INB) and creating additional validation queries are both good techniques for verifying a query definition.

For reference, validation tables can be found in Appendix B.
Navigating Between Sections

Hyperion organizes document definition and content into sections. The Query section contains the data model and query definition. Meanwhile, the Results section contains the queries result set. If the query result set viewed in the Results section does not appear to be correct, toggle back to the Query section by clicking the section name in the Section Catalog.

Another option to toggle between sections is to use the blue arrows in the upper right-hand corner of the application.

6. Click Query in the Section Catalog or the toggle button to go back to the Query Section.

Canceling a Query

Canceling a processing query will send a command to the database to discontinue query execution. In order to proactively cancel a processing query, simultaneously press ALT + END and hold until the cancel message is received.
Chapter 6: Limiting the Query

Limits narrow a query result set so that it only includes the data required to answer the business question. Limits are based on one or more logical conditions consisting of columns, operators and values. When specifying a limit, data is retrieved from the database if, and only if, it meets the specified limit conditions.

Section 1 – Applying a Limit

The sample business question asked to retrieve data for this semester only. The semester field SFRSTCR_TERM_CODE will be used to create this limit.

1. Launch the Limit dialog box by doing one of the following to the SFRSTCR_TERM_CODE column:
   - Double-click the column;
   - Right-click the column and select Limit; or
   - Drag and drop the column to the limit line.
Limit Dialog Window

The name field includes a custom name for the specific condition built using the dialog. This field defaults to the column on which the condition is built. This value may be changed to a meaningful, descriptive value in order to increase query readability and intuitiveness.

The Operator Selection Dropdown includes a list of valid comparison operators that may be used to define a condition. These operators will be used to compare each column or calculation to one or many values.

- **Equal** - retrieves only records where the limited column equals the specified value(s).
- **Not Equal** - retrieves only records where the limited column does not equal the specified value(s).
- **Less Than** - retrieves only records where the limited column is less than the specified value(s).
- **Less or Equal** - retrieves only records where the limited column equals, or is less than, the specified value.
- **Greater Than** - retrieves only records where the limited column is greater than the specified value(s).
- **Greater or Equal** - retrieves only records where the limited column equals, or is greater than, the specified value.
- **Begins With** - retrieves only records where the limited column begins with the specified value(s) up to and including the end value.
- **Contains** - retrieves only records where the limited column contains the specified value(s) regardless of location.
- **Ends With** - retrieves only records where the limited column ends with the specified value(s).
- **Like** - retrieves records where a text string appears and reflects the placement of the specified value(s). For example, Name like %Ze_ would retrieve records for all employees whose name contains Ze followed by a single character at the end.
- **Between** - retrieves only records where the value of the limited column lies between or includes the specified values.
- **IsNull** - retrieves only records where the limited column has no value. For example, a field in which no data has been entered.

The Free-Form Limit Value field will be used to enter a value for the condition. This field accepts free-form entered values. It will validate and add those values to the value list when the green check button is clicked.
The **Show Values** button will generate a pick list of distinct, valid values for the column on which the condition is being built.

**NOTE:** Proceed with caution when using **Show Values**. This option may be time consuming if the column in question contains a large number of records.

The **List of Condition Values** field will contain the list of values that will be compared to the column or calculation using the selected operator. This list may be populated by free-form entry using the **Custom Value** field or by selecting from a list of valid values using the **Show Values** button.

These two options for populating this list are **mutually exclusive**. If the list has been populated using **Show Values** but instead must be populated using free-form entry, click **Custom Values** to reset the *Limit* dialog box for free-form entry. If free-form entry values have been entered but a list of valid values is desired, then click **Show Values**.

2. Click **Show Values** and select 200780. The limited value list should appear as shown to the right:

Multiple values may be selected when building a condition by pressing the CTRL key while clicking on the value. Pressing the **SHIFT** key may be used to select a contiguous list of values in the pick list.

3. Click **OK** to add the condition to the **Limit** line. The **Limit** line should appear as shown to the right.

If you do not see the **Limit** line, it may be hidden. To unhide it, go to the upper corner of the Request line, and click on the **Limits** link; the **Limit** line will then appear under the Request line as seen in the above screen shot.
NOTE: Hovering over an item on the limit line will display the limits that have been applied to the topic item.

4. Process the query.

Section 2 - Adding Additional Limits

Though the result set has been limited to the current semester, it can be limited additionally by the registration status code SFRSTCR_RSTS_CODE should you not want to include audits, drops, and withdraws.

5. Launch the Limit dialog box for the SFRSTCR_RSTS_CODE column.

6. Click Show Values to display the values for the column.

   Though you could use CTRL to select the multiple values that begin with ‘R’, if you look at STVRSTS, the Registration Status Validation table, you would notice that there are additional registration status codes that begin with ‘R’.

   Instead of this, go to step 7...

7. Using the Custom Values button:
   A) select the Custom Values button
   B) change the Operator Selection Dropdown to Begins With
   C) add R to the Free-Form Limit Value field
   D) click OK
The limit line should appear as follows after adding the new conditions:

![Limit Line Example]

By default, Hyperion will separate multiple conditions on a limit line with AND logical operators. The AND logical operator signifies that the result set will be constrained by all conditions. These operators may be changed to the OR logical operator by double-clicking on the operator itself. The OR logical operator signifies that the query will retrieve all records that meet condition 1 OR condition 2 OR condition 3...

The conditions will be performed in the order that they appear on the limit line. This is especially to note when using the OR logical operators.

8. **Process** the query.

**Removing Items from the Limit Line**

Items may be removed from the limit line by doing the following:

- **Individual item removal**: Click to select the condition on the limit line, right-click and select **remove**.
- **All item removal**: Highlight the word **limit** on the limit line, right-click and select **remove**.
In Class Exercise – Chapter 6

1. Launch the Limit dialog window for SFRSTCR_ADD_DATE. Check dates since July 1st. Process the query. Record how this differs from the original results.

2. Remove SFRSTCR_ADD_DATE from the limit line and launch the Limit dialog window for SFRSTCR_ACTIVITY_DATE. Check dates since July 1st. Process the query. Record how this differs from the add date results.

3. Remove SFRSTCR_ADD_DATE AND SFRSTCR_ACTIVITY_DATE from the limit line.

Answers:
1. Select the Custom Values button, change the Operator Selection Dropdown to >= Greater or Equal, and add 7/1/2xxx to the Free-Form Limit Value Field. The result rows should be considerably less.
2. CTRL-Click on SFRSTCR_ADD_DATE AND SFRSTCR_ACTIVITY_DATE to select the conditions then the Delete key.
3. Click to select the condition on the limit line, right-click and select remove.

Chapter 7: Aggregating the Query

**Section 1 – Define Calculations**

In addition to table columns, the request line may also include calculations. Calculations apply mathematical operations to a table column in order to derive totals in the result set.

Recall that the sample business question asks for a count of the registered.

| This counting operation requires adding a calculation to the Hyperion request line. This calculation will count each individual person grouped by term and gender values. |

The calculation functions available in Hyperion include:

- **Sum** - performs a sum of all values included in the column. In order to sum column values, the column must be a numeric data type.
- **Average** - returns the average of all values in a column. In order to average column values, the column must be a numeric data type.
- **Maximum** - returns the maximum value for all values within a column.
- **Minimum** - returns the minimum value for all values within a column.
- **Count** - counts each occurrence of data within a column. This operation is equivalent to counting the number of records that meet a query’s condition criteria.
- **Count Distinct** – counts the distinct occurrences of data values within a column. This operation avoids double counting duplicate values within the column.

**Aggregation Behavior**

Notice that each of these calculation functions makes the most sense when performed on many data values over many database records. For example, the sum function applied to one value equals that value. The sum function applied to ten values equals a much more meaningful condensed total of all ten values.

An operation that summarizes and consolidates records to perform a calculation is called an **aggregation** operation.

Aggregation operations summarize many data values using the defined calculation function to produce a total result. The following diagram depicts the aggregation behavior.
NOTE: The next two diagrams are not using data in relation to the business question example. Instead, they will draw from currency examples and the business question will pick up again in Adding an Aggregate Calculation to a Query.

### Level of Aggregation

When an aggregation operation is performed alongside other columns in a query request line, those columns dictate the level of aggregation for the operation. A query that includes aggregation will calculate an aggregate result for every distinct combination of values in the level of aggregation columns. The following diagram depicts the level of aggregation behavior.

Understanding the impacts of aggregation on a query is very important when retrieving a calculated result.

**Section 2 - Adding an Aggregate Calculation to the Query**

1. The query in its current state displays the detail registration records for each unique internal ID or PIDM. Before a count can be added, columns that provide data distinct to each PIDM must first be removed.
Within the Request line, **CTRL-Click** SFRSTCR_CRN, SFRSTCR_PTRM_CODE, SFRSTCR_RSTS_CODE, SFRSTCR_BILL_HR, SFRSTCR_ADD_DATE, and SFRSTCR_ACTIVITY_DATE and click the **Delete** key.

You will receive a confirmation message, click **Remove**.

2. Right-click the SFRSTCR_PIDM column on the request line, select **Data Functions > Count Distinct**.

After **Count Distinct** has been selected, the Request line will display as:
If a Count had been performed on the Pidm instead of the Count Distinct, the result would be that the student would be counted for each class that she/he is registered.

3. **Process** the query and view the results.

Below is an example of results, the actual counts may vary during the class.

<table>
<thead>
<tr>
<th>Sfrscr Term Code</th>
<th>Sfrscr Pidm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200780</td>
</tr>
<tr>
<td></td>
<td>15332</td>
</tr>
</tbody>
</table>
In Class Exercise – Chapter 7

1. Count the number of courses registered. Compare to the number of students.

__________________________________________________________________

2. Sum the number of billable hours registered. Compare to the number of students and courses.

__________________________________________________________________

Answers:

1. Add the SFRSTCR_CRN to the request line; Right-click; Data Functions, Count. Process the query and compare the results.
Chapter 8: Creating Join Relationships

Section 1 – Building a Join

Table relationships must be created when the information required to answer a business question resides in multiple tables. These relationships are referred to as joins. They consist of one or many common columns between two tables and an operator that will be used to compare the values within the common columns in order to identify related records.

Recall that the business question requests counts by gender.

From the class Exercises, we know that SPBPERS contains the gender or sex of a student. We will now include this in the document data model, and create the join to SFRSTCR.

Adding Additional Tables to the Content Pane

1. Return to the Query section of the document by toggling or selecting the section. Add the SPBPERS table to the content pane by identifying it within the catalog and dragging it into the data model.
2. Drag and drop the **SSFRSTCR_PIDM** from the SFRSTCR table to the **SPBPERS_PIDM** in the SPBPERS table. This creates the equality relationship between the two tables.

After the drag and drop a line will appear between the two fields:

3. Add the **SPBPERS_SEX** column from the SPBPERS table to the request line after the term code and before the count.

4. **Process** the query and view the results.

Below is an example of results, the actual counts may vary during the class.
Section 2 - Auto-Joins

In some instances of Hyperion, join relationships between tables will automatically be created as soon as the second table is added to the data model. This default behavior, called auto join, automatically creates equality join relationships between all columns within the two tables that share the same name and same data type. Unfortunately, this behavior often produces erroneous join relationships and contributes to incorrect results.

Below is an example of this behavior.

The Auto Join option should be turned off during the first Hyperion query session that requires multiple tables. After turning auto join tables off once, all future data models will not demonstrate this behavior.
Removing Auto Join Tables

Turning the auto join default behavior off:

5. Return to the Query section of the document by toggling or selecting the section.

In the toolbar, select Data Model > Data Model Options.

The Data Model Options dialog box will appear.

6. Uncheck Auto join tables under the General tab. Click OK.

7. Remove the second table and add it again. No joins will be present this time. Proceed by manually adding the join(s).
Section 3 - Join Rules and Validation

Consider the following rules when creating a join between two tables:

- Understand the numeric relationships between the tables (e.g., one student may have registered for many courses).
- Identify the combination of columns that define uniqueness for each table in the list.
- Identify the common columns between each table in the list and ensure that at least one of the tables will use its primary key columns to define a relationship.
- Apply an equality relationship between these common columns in order to link records between tables.

The visual joins reflected in the Hyperion data model represent real relationships that will be enforced by the query when it is passed to the database. This visualization acts as a tool to assist with join validation and verification when building a new query.

**Cartesian Join**

If joins do not exist between two tables that both contribute columns to the request line, then the database will have no means to compare the records between the two tables. This behavior results in a Cartesian Join between the two tables, which retrieves a result set equaling the total number of records from one table multiplied by the total number of records from the other table.

This situation should be avoided at all costs because it produces a very large result set consisting of meaningless results and it consumes unnecessary resources on the database server that will affect other Hyperion users.

Removing Joins between Tables

Joins between tables may be removed by doing either of the following:

- Click on the join line, right-click and select Remove.
- Click on the join line; press Delete on the keyboard.
In Class Exercise – Chapter 8

1. Add the term description to the request line. (Hint: STVTERM table). What does 200780 represent?

__________________________________________________________________

Answers:
Chapter 9: Sorting the Results

Sorting organizes a result set to promote readability and data analysis. Typically, this involves rearranging the result set based on an ascending or descending alpha-numeric order applied to one or many selected columns or calculations. When many columns/calculation are included in a sort definition, Hyperion will organize the data based on a combined hierarchical order that includes the first column/calculation sort, the second column/calculation sort, the third column/calculation sort and so on. This is called a nested sort.

Applying a Sort

The result set in the example Hyperion document does not include a defined sort. The result set would be more legible if a sort order was applied to the columns.

1. Click on the Query label in the Section Catalog to navigate back to the query section.

2. In order to define a sort that organizes the result set in ascending alpha-numeric order by Gender values, drag the Spbpers Sex column from the Request line and drop it on the sort line.

If you do not see the Sort line, it may be hidden. To unhide it, go to the upper corner of the Request line, and click on the Sort link; the Sort line will then appear under the Request and Limit lines seen in the above screen shot.

After dragging and dropping the Spbpers Sex column to the Sort line, it will appear as:
NOTE: The column must be dragged from the request line. A column has to exist in the result set before it may be added to a sort definition.

**Sort Ascending or Descending**

A column or calculation may be changed to sort either ascending or descending. The Hyperion default is ascending. In order to change the sort order, use one of the following methods:

- Double-click the column or calculation on the sort line. This toggles the arrow head direction on the column or calculation and flips the sort order.
- Select the column or calculation on the Sort line and click one of the sort buttons on the toolbar.

**Removing Items from the Sort Line**

Items may be removed from the sort line by doing the following:

- Individual item removal: Click to select the column on the sort line, right-click and select Remove.
- All item removal: Highlight the word Sort on the sort line, right-click and select Remove.
Chapter 10: Formatting the Results

After finalizing the query definition and validating the results, a few simple formatting steps might increase the readability and usability for any end consumer of the data. These simple formatting techniques include:

- Moving Columns
- Rename Result Columns
- Formatting Numbers
- Resizing Result Columns
- Hiding the Section/Catalog Pane
- Adding Grand Totals to Columns
- Aligning Data

Section 1 – Moving Columns

In some cases, the result set column order may not match the requirements for the business problem. Column positions may be altered in the results section by using one of the following methods:

- Click and drag the column name in the Outliner to the new location
- Click and drag the column directly in the result set

Generally, the outliner method is preferred when the result set contains several columns.

1. Click on the Results label in the Section Catalog to navigate back to the Results section.
2. In the **outliner**, click and hold **Stvterm Desc** until the cursor appears then **drag** the column name to the right and **drop** it after **Sfrstcr Term Code**.

After the drag and drop, the outliner should display as:

![Outliner](image)

**Section 2 – Rename Result Columns**

In some cases the columns in a query are not named using exact business terminology. These column names transfer directly to the **column headers** in the results section. When applying an aggregate calculation to a column, the name of the column on which the aggregate operation is performed transfers to the column header in the result section. In both cases, the **default column headers created by Hyperion within a result set may not match the business requirements for the report**. The column headers may be changed by doing the following:

3. Navigate back to the **Query** section and within the Request line, right-click on **COUNT(Sfrstcr Pidm) > Properties**.

The **Item Properties** dialog box will appear.
4. In the *Item Properties* box, enter **Student Count** within the name field, click **OK**.

![Item Properties dialog box](image)

5. **Process** the query and navigate back to the results section in order to view the new column header.

![Student Count header](image)

6. Navigate back to the **Query** section and change the name of **Count(Sfrstcr Crn)** to **Course Count** and **SUM(Sfrstcr Bill Hr)** to **Billable Hours**.

**Process** the query and navigate back to the results section in order to view the new column headers.

When finished the results will display as (numbers may vary):

![Table with results](image)
Section 3 – Formatting Numbers

Properly formatting numbers within a result set can make the results easier to read and understand. The columns containing numbers may be changed by doing the following:

7. Navigate back to the Results section and right-click on the Student Count column > Number.

The Properties dialog box will appear.

8. In the Properties box > Format area, select #,##0 and click OK; the box will disappear and the Student Count column will now be formatted in this way.
9. Perform steps 7-8 for Course Count and Billable Hours columns.

Section 4 – Resizing Result Columns

Result set columns may be resized by doing the following:

- Hover the cursor over the column’s right border until a double arrow appears. Click in order to bring focus to the double arrow and manually resize the column left or right.
- Hover the cursor over the column’s right border until a double arrow appears. Double-click in order to automatically resize the column so that the longest column value will be visible

10. Use the automatic resize technique defined above in order to resize the Spbpers Sex column in the result set.

11. Repeat this process for each column with the Results set.
Section 5 – Hiding the Section/Catalog Pane

In order to create additional workspace in the Content Pane for result set formatting, the Section and Catalog Panes may be hidden. In order to hide the Section and Catalog Panes:

12. Within the toolbar, click the **Show Section/Catalog**. This button toggles the Section and Catalog panes between visible and invisible.

Section 6 – Suppressing Duplicates

In many scenarios, column values will repeat in the result set after a sort order has been applied. In queries that include aggregate calculations, this behavior often occurs on columns that exist within the level of aggregation list (specifically, the columns that are not at the lowest level of aggregation). The result set may be easier to consume if the duplicate values are suppressed.

In order to suppress duplicates within a column:

13. Highlight the **Stvterm Desc** column by clicking within the column data; the column will turn black to indicate that it has been highlighted.

Right-click and select **Suppress Duplicates** from the fly-out shortcut menu; the duplicates will disappear.
14. Repeat step 13 for the **Sfrstcr Term Code** column.

After completing the **Suppress Duplicates** step for both columns, they should display as:

<table>
<thead>
<tr>
<th>STerm Desc</th>
<th>Sfrstcr Term Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall Term 2007</td>
<td>200700</td>
</tr>
</tbody>
</table>

**Section 7 – Adding Grand Totals to Columns**

A grand total summarizes all values in a numeric column. In order to summarize the count of students:

15. Highlight the **Student Count** column by clicking within the column data. The column will turn black to indicate that it has been highlighted.

   Right-click and select **Grand Total...** from the fly-out shortcut menu.

   The **Insert Grand Total** dialog box will appear.

16. With **Student Count** highlighted, **Ctrl-Click** on **Course Count** and **Billable Hours**, click **OK**.
Scroll to the bottom of the result set to verify the addition of a grand total amount.

<table>
<thead>
<tr>
<th>SBTN Term Desc</th>
<th>SBTN Term Code</th>
<th>SbPers Sex</th>
<th>Student Count</th>
<th>Course Count</th>
<th>Billable Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall Term 2007</td>
<td>200780</td>
<td>F</td>
<td>10011</td>
<td>41689</td>
<td>117676</td>
</tr>
<tr>
<td>Fall Term 2007</td>
<td>200780</td>
<td>M</td>
<td>5943</td>
<td>24321</td>
<td>67871</td>
</tr>
<tr>
<td>Fall Term 2007</td>
<td>200780</td>
<td>N</td>
<td>14</td>
<td>56</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>15668</strong></td>
<td><strong>65966</strong></td>
<td><strong>185801</strong></td>
</tr>
</tbody>
</table>

This function, used to produce the grand total, may be changed by double-clicking on the grand total amount and selecting a new Grand Total Function within the Modify Total Function window.

**Adding Text to the Grand Total Line**

Adding a text-based label identifies the contents in the grand total cell. Free-form text may be inserted into any cell produced on the grand total row by doing the following:

17. Double-click the empty cell under the SpbPers Sex column on the grand total row.

18. In the Custom Function dialog box, type ‘Grand Total’, click OK; the Modify Total Function dialog box will appear.
19. In the Modify Total Function box, click OK.

![Modify Total Function dialog box]

20. The Grand Total label appears to the left of the grand total amount.

<table>
<thead>
<tr>
<th>Code</th>
<th>Shtpers Sex</th>
<th>Student Count</th>
<th>Course Count</th>
<th>Billable Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td></td>
<td>10011</td>
<td>41589</td>
<td>117676</td>
</tr>
<tr>
<td>M</td>
<td></td>
<td>5643</td>
<td>24321</td>
<td>67571</td>
</tr>
<tr>
<td>N</td>
<td></td>
<td>14</td>
<td>56</td>
<td>154</td>
</tr>
<tr>
<td>Grand Total</td>
<td></td>
<td>15668</td>
<td>65966</td>
<td>185801</td>
</tr>
</tbody>
</table>

**Removing Grand Totals**

In order to remove a grand total amount, highlight the row by clicking on the row number. Then, right-click and select Remove Row from the fly-out window.

![Remove Row option]

**Section 8 – Adding Subtotals**

A Break Total is a subtotal that may be added to a result set column.

The final exercise will take you through the steps of creating a break total.
Section 9 – Aligning Data

Aligning data within a result set can also make the results easier to read and understand. Alignment can be changed by doing the following:

21. In the Results section, Highlight the Spbpers Sex column by clicking within the column data. The column will turn black to indicate that it has been highlighted.

   Right-click and select Alignment from the fly-out shortcut menu; the Properties dialog box will appear.

22. In the Properties box > under Horizontal Alignment, select Center; the Preview pane will reflect your choice. Leave all other options at their default.

   Click OK.

The Spbpers Sex column will then display as:
Chapter 11: Managing Hyperion Documents

Section 1 – Exporting the Result Set

After processing a query, the data from the results section may be exported to a variety of different file formats. The two most common supported export formats are .XLS and .PDF. In order to export the results of a query:

1. Navigate to the results section and from the toolbar select File > Export > Section; the Export Section dialog box will appear.

2. In the Export Section box, navigate to the appropriate folder location, enter a file name, select the export file type and click Save.
Section 2 – Inactivating Save Query Results from the Document

Retaining the Saved Query Results in the document can cause confusion for other users when they open the document. In order to ensure that one has the most current data each time, the Save Query Results with Document option needs to be inactivated. This is especially true before requesting that a report be published in the Hyperion Workspace.

3. In the toolbar, click File > Save Options > Save Query Results with Document; the Save Query Results with Document dialog box will appear.

4. Within the Save Query... box, under Query Results, Query will have an X in the box next to it. Click on the X to deselect Query. Click OK.

5. Save the document using File > Save As on the toolbar.

Each time the document is opened from this point forward, the results will be blank. One must select Process in order to have the results populated.
Appendix A – Final Exercise

For the final exercise, login to Hyperion and find the *Enrolled by College and Gender.bqy* under the *Spectrum Training* folder. Save this document to the desktop.

With the exception of the number of courses and the billable hours, *Qry1 Registration Count by*... requests the same fields as the *Enrolled by Gender.bqy* from the previous exercises.

There is one notable difference though in that a new table SGBSTDN has been added and joined. We will use the SGBSTDN in the exercise. Please note though that the data in the SGBSTDN does not get updated each semester that a student is here. We must therefore find the most current record for the student in which the student record term does not exceed the registration term. This has already been done for you within this exercise by the creation of a subquery.

Our original business question instructed us to count the enrolled students by gender. The list included those registered at the U of M for this semester. It also was sorted by gender.

**Expanded Business Question**

We now need to count the enrolled students by college in addition to the gender. The list should include only registered new first time freshman. Sort these results by college and gender. Calculate a Subtotal number of students by College. Calculate a Grand Total number of students.

1. Bring the STVSTYP, Student Type Validation table, to the data model and find the student type code that corresponds to new first time freshman.

2. After finding the student type, remove the STVSTYP table from the data model pane and limit the query by the student type in SGBSTDN.

3. Add the name of the college the student is assigned to at the U of M by joining to the STVCOLL, College Description Validation table.

4. Rename each of the fields within the request line respectively to be Term Code, Term Desc, Gender, Number of Students, and College.

5. Process the query.

6. The Expanded Business Question requires the sort to be by College and Gender. Create this sort.

7. Expand and/or contract the Term Code, Term Desc, College, and Gender column in the result set to the field’s maximum width. Specifically, contract the field width of Number of Students so that Students falls on the line below Number of.
8. Center the Gender columns.

9. Add a **grand total** and **description** for the Number of Students.

10. Add a **break total** for the number of students by College by right-clicking on the College column and selecting **Break Total**... The Insert Break Total dialog window displays. Select the following for each section:
    
    * At every break in: College
    * Break total function: Sum
    * Add break total to: Number of Students

    Click OK.

11. Remove the break total description from the display.

12. **Suppress Duplicates** for Term Code, Term Desc, and College.
Final Exercise Answers

1. Expand the table catalog pane and drag and drop the STVSTYP to the data model pane. Right click and select Detail View.

2. Right click STVSTYP and select Remove. Double-click the Sgbstdn Styp Code and enter N in the free form box then select OK.

3. Expand the table catalog pane and drag and drop the STVCOLL to the data model pane. Join SGBSTDN to STVCOLL by dragging and dropping the sgbstdn coll code 1 to stvcoll code. Drag and drop stvcoll desc to the Request line.

4. Double-click Sfrstcr Term Code or right-click and select Properties. In the name field change it to be Term Code. Click OK.
   
   Repeat steps for each:
   
   Sfstterm Desc = Term Desc
   Sfppners Sex = Gender
   COUNT(Sfrstcr Pidm) = Number of Students
   Sstvcoll Desc = College

5. Click on the Process button in the toolbar.

6. Drag and Drop College from the request line to the sort line. Move gender to be to the right of college. Process the query. Move the College to the left of the gender in the result set.

7. For the first four fields, hover over the lines that separate each column and double-click. For Number of Students, hover over the right horizontal in the Number of Students column until the arrows display. Click and drag the line to the left and release. Repeat until Students wraps under the Number of.

8. Right click on the column. Select Alignment... from the shortcut menu. Select center under Horizontal Alignment within the Properties Alignment dialog window. Click OK.

9. Right-click on the Number of Students column. Select Grand Total... from the shortcut menu. On the Insert Grand Total dialog window under Add grand total to: select Number of Students. Click OK. Double-click in the blank box beside the Grand Total that was added. Within the Custom Function dialog window add ‘Total’. Include the single quotes. Click OK. The Modify Total Function dialog window displays. Click OK.

10. N/A

11. Double-click on one of the Total descriptions under College. The Modify Label dialog box appears with "Total + ToChar(College)." Delete it from the box and click OK.

12. Ctrl-Click Term Code, Term Desc, and College. Right-Click and select Suppress Duplicates from the shortcut menu.

   The final result will look like the table to the right.

   *The totals will vary at class time
## Appendix B – Tables

### Student Module Base Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Description</th>
<th>Sample Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBBABUD</td>
<td>Applicant Budget Table</td>
<td>Budget Type</td>
</tr>
<tr>
<td>RBRACMP</td>
<td>Applicant Budget Component Table</td>
<td>Component Type, Amount</td>
</tr>
<tr>
<td>RCRAPP1,2,3...</td>
<td>Financial Aid Application Tables</td>
<td>EFC</td>
</tr>
<tr>
<td>RFRBASE</td>
<td>Fund Base Data Table</td>
<td>Fund Titles, Source</td>
</tr>
<tr>
<td>RORSTAT</td>
<td>Applicant Status Base Table</td>
<td>Budget Group</td>
</tr>
<tr>
<td>RPRAWRD</td>
<td>Applicant Award Table</td>
<td>Fund Code, Accept Amount, Decline Amount, Cancel Amount</td>
</tr>
<tr>
<td>SARADAP</td>
<td>Admissions Applicant Repeating Table</td>
<td>Application Date, Student Type, Recruiter, Rate, Intended College / Major, Withdraw Reason</td>
</tr>
<tr>
<td>SARAPPD</td>
<td>Student Application Decision Repeating Table</td>
<td>APDC Codes</td>
</tr>
<tr>
<td>SARCHKL</td>
<td>Admissions Checklist Repeating Table</td>
<td>Checklist Items and Dates</td>
</tr>
<tr>
<td>SFBETRM</td>
<td>Student Registration Table</td>
<td>Eligibility Status Code, AR Indicator</td>
</tr>
<tr>
<td>SFRSTCR</td>
<td>Student Course Registration Repeating Table</td>
<td>CRN, Part-of-term, Credit Hours, Registration Status Code</td>
</tr>
<tr>
<td>SGBSTDN</td>
<td>Student Base Table</td>
<td>Level, Type, College, Degree, Major, Rate</td>
</tr>
<tr>
<td>SGRADVR</td>
<td>Multiple Advisors Repeating Table</td>
<td>Advisor Code, Description, Primary Indicator</td>
</tr>
<tr>
<td>SHRTGPA</td>
<td>Term GPA Table</td>
<td>Hours Attempted, Earned, Quality Points, GPA</td>
</tr>
<tr>
<td>SOBSBG1</td>
<td>Source / Background Institution Table</td>
<td>Sources, High Schools, College/University</td>
</tr>
<tr>
<td>SORFOLK</td>
<td>Parent / Guardian Information Repeating Table</td>
<td>Parent Last Name, First Name, Relationship</td>
</tr>
<tr>
<td>SORHSCH</td>
<td>Person Related High School Repeating Table</td>
<td>High School Attended</td>
</tr>
<tr>
<td>SORPCOL</td>
<td>Person Related Prior College Repeating Table</td>
<td>Prior Colleges Attended</td>
</tr>
<tr>
<td>SORTTEST</td>
<td>Test Score Repeating Table</td>
<td>Test Code, Score, Date</td>
</tr>
<tr>
<td>SRBRECR</td>
<td>Recruiting Base Table</td>
<td>Recruit Type, Recruiter, Major, College, Withdraw Reason</td>
</tr>
<tr>
<td>SSBSECT</td>
<td>Section General Information Base Table</td>
<td>Course Number, Title</td>
</tr>
<tr>
<td>SSRMEET</td>
<td>Section Meeting Times Repeating Table</td>
<td>Begin/End Times, Building, Room</td>
</tr>
</tbody>
</table>
## Validation Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Table Description</th>
<th>Sample Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>STVCNTY</td>
<td>County Validation Table</td>
<td>County Description</td>
</tr>
<tr>
<td>STVCOLL</td>
<td>College Validation Table</td>
<td>College Description</td>
</tr>
<tr>
<td>STVDEGR</td>
<td>Degree Validation Table</td>
<td>Degree Description</td>
</tr>
<tr>
<td>STVDEPT</td>
<td>Department Validation Table</td>
<td>Department Description</td>
</tr>
<tr>
<td>STVETHN</td>
<td>Ethnicity Validation Table</td>
<td>Ethnicity Description</td>
</tr>
<tr>
<td>STVMAJR</td>
<td>Major Validation Table</td>
<td>Major Description</td>
</tr>
<tr>
<td>STVNATN</td>
<td>Nation Validation Table</td>
<td>Nation Description</td>
</tr>
<tr>
<td>STVTERM</td>
<td>Term Validation Table</td>
<td>Term Description</td>
</tr>
</tbody>
</table>