PHR Data Warehouse Training
This training manual is intended for Hyperion (Brio) client users with access to the PHR (Payroll Human Resources) Data Warehouse tables. The information provided will assist users in understanding commonly used tables, data elements, joins, limits and computed items when using PHR data warehouse tables. Sample queries and training exercises will reinforce proper use of the PHR data elements. Reference tools are also provided.

**Brio Client vs. WOW (or Quickview) access** – Users are sometimes defined by the type of software they are using to access data in the Data Warehouse.

Brio Client access refers to users who use the Brio Intelligence Explorer software. The software is installed on a user's workstation and they can see a list of tables to which they have access. This software offers the greatest flexibility in creating, modifying and processing ad hoc queries and reports. All queries and reports made available on WOW, were developed by Brio Client users. This is licensed software which may be purchased from OIT Software Licensing and costs $660 per workstation.

WOW or Quickview access refers to users who use the Brio Quickview web plug-in software. The software is installed on a user's workstation for use with a web browser such as Netscape or Internet Explorer. Users do not see a list of tables to which they have access as the Client users do, because they cannot create or modify queries. Brio Quickview provides the greatest of ease in its use, because a user only needs to click a button or two to automatically process a "canned" query to which they have access to produce a report. Quickview users can also change limits within a query if prompted, and export and locally print results data or reports. This is licensed software which may be requested from the Office of Data Administration and is free of charge.

**PHR Data Warehouse Meta Data** – Each field in the PHR tables, which are referred to in this document, have had definitions written and stored for your access under the **Show Remarks** function in the Query section. We strongly encourage you to familiarize yourselves with this Brio feature and with the meanings of each of the PHR fields. Accurate queries and correct reports require a firm understanding of the information conveyed in the various data stored in the PHR data warehouse. Study, thought, trial and error are all requirements in becoming a seasoned Brio reporter.

**PHR Data Refresh Cycles** - After data is initially loaded into the data warehouse, they must be refreshed on a periodic basis. The PHR Data Warehouse tables are refreshed daily Mon-Friday by 9 a.m. with the prior days updates and additions.

It is important to be aware of the refresh cycles when running a query. The difference between the actual transaction system and the data warehouse refresh cycle can affect the results obtained in a
query. Refresh dates can be checked for each data subset by checking the **Extract Date** associated with each record within a table.

**PHR Employee History** – The PHR (Payroll Human Resources) System was implemented in a phased-in method at the University of Maryland, College Park beginning July 2001 and completed by mid November 2001. Only the current employment record for all active employees was entered into the PHR application/tables. If an employee was not active at the time of conversion, there was not a PHR record entered for them. There were no additional historical records converted into this system.

**PHR Institutions** - There are 5 USM Institutions that are currently using the PHR (Payroll Human Resources) System. They include:

- University of Maryland, College Park – (UMCP - Inst Code 01)
- University of Maryland Biotechnology Institute (UMBI – Inst Code 03)
- University of Maryland, Eastern Shore (UMES – Inst Code 04)
- University of Maryland, Center for Environmental Science (UMCES – Inst Code 07)
- University of Maryland, Systems Office (USMO – Inst Code 08)

**PHR Data** includes current information on employee and appointments. The Employee tables will store that last current employee record (even when an employee terms, that last record will remain in the current employee table). The appointment tables behave a little different from the employee tables in that, once the appointment terminates, the appointment information will move to history and not appear in the Data Warehouse current tables.

*Please Note: We are in the process of reviewing Employee and Appointment CFH tables that contain Current, Future and History information all in one table. In the near future, we hope to push these tables out to Brio Client users, as well as update some of our current WOW reports to incorporate history information where appropriate.*

**PHR_DV_Employee** and **PHR_DV_Appt** are two ‘data marts’ that have been built to eliminate the need for many of the joins you would otherwise have to do if you were trying to draw data from the PHR ‘home tables’. Basically, a ‘data mart’ is a data warehouse table that is made up of fields drawn from a collection of other data warehouse tables. It is used to combine many related elements into one, user-friendly place.
**PHR Point in Time “PIT” data** is available to assist you with capturing PHR data at a particular Point in Time (PIT). The PHR_EDV_PIT table began loading in January 2005. This table freezes a copy of all the data elements that are stored in the PHR_EDV table. Critical employee and appointment information are provided in this table including salary and funding details.

This table was created to provide client users with the capability to produce data reports based on prior time periods. This is extremely useful when comparing statistics from month-to-month or year-to-year. For example, if you wanted to compare Employee Counts or Salaries within your units from month-to-month, you could produce these stats using this table. Another use of this table would be to identify employees that were paid from Funding Accounts during a particular month(s). This table will also be very beneficial when we freeze the data in June (just prior to fiscal year changes), so that you can compare salaries or funding changes before and after the fiscal year.

*The PHR EDV PIT table is loaded every pay period,* with the current employee and appointment data at that particular extract date. This data will stay frozen in the table. It is important to note that when using this table, there are 3 additional fields that will assist you with your point in time report processing. The additional fields include *"Fiscal Yr", "Month", and "Fypper".* These fields will be populated with the Fiscal Yr (4 digits - ex. 2005), Month (2 digits - ex. 03), and FYPPER (Fiscal year and 2-digit pay period combination - ex. 200515) at the time the data is frozen. Depending on the "PIT" that you are seeking, you will most likely want to limit on any combination of these fields.
Where to Get PHR Data Warehouse Help

Who to Contact

Kim Colbert   x55664   kcolbert@umd.edu
Natalie Torres   x55674   ntorres@umd.edu

Other Helpful References

1. Data Definition Search

The Data Administration Data Definition Search Page is located:  http://dad.umd.edu/data-element-search/

2. Meta Data Definition Report

The Meta Data Definition Report is located on WOW under the Data Warehouse Reports folder. This report generates a list of data definitions based on the Data Warehouse tables that you select.

3. PHR Home Page

The PHR Home Page has many resourceful documents, including the PHR Business Rules and PHR Glossary. This site is located:  https://uhr.umd.edu/phr/
Commonly Used PHR Tables/Data Elements

**Phr_Dv_Employee** – This data mart combines information from multiple employee tables in the Data Warehouse, thus eliminating the need for multiple joins in most cases. There will only be one record per employee in this table, which is secured based upon your unit.

The following critical data are stored in this table:

- **Active_Empl_Ind** – Active Employee Indicator is a Y/N flag which indicates if the employee is currently, actively employed or not. This field is set to Y (Yes) in the current DW extract if: Empl_Status_Cd is equal to:
  - A – Active
  - N – Returning AA (academic year appt)

  Otherwise, in all other cases, Active Employee Indicator is set to N (No).

- **Empl_Status & Empl_Status_Cd** – These fields describe the current status of the employee at the time of the load of PHR employee data to the DW. The codes include:
  - A – Active
  - N – Returning AA (academic year appt)
  - F – Future
  - I – Inactive
  - Q – Non-Paid
  - E – Emeriti Only
  - T – Terminated w/ a Future Appt
  - X – Incomplete

- **Address** *Note: for business addresses, an employee will only have on-campus fields populated OR off-campus fields populated.
  - **Business – On Campus:**
    - Room
    - Ups_Building
    - Email_Address
  - **Business – Off Campus:**
    - Addr_Street1
    - Addr_Street2
    - Addr_Street3
    - Addr_City
    - Addr_State
    - Addr_Zip
* Note: This table does not yet retain data on employees after the Pay Period in which they separated from the University has passed. We are in the process of adding this (and future-related) information, and will let you know when it is available for your use.
**PHR_DV_EMPLOYEE**

**Exercise 1: Review Phr_Dv_Employee Table**

1) **Open** the Brio Client.

2) From the Section/Catalog, **Open the Phr_Dv_Employee** table.

3) **Drag** the Employee Data Elements to the Request Line.

4) **Limit** on your Unit_Cd

5) **Click on Process** and **Review** your results.
**Phr_Dv_Appt** – This data mart is a secured table storing data for employees with a current approved appointment in your unit. Included are critical appointment data extracted from multiple appointment tables in one record for each current appointment the employee holds. Using this table can help eliminate the need for multiple table joins.

The following critical data are stored in this table:

- **Phr_Appt_Internal_Id** – PHR Appointment Internal Id is a unique number which represents the Internal Identification for the appointment of the associated employee. This is a randomly generated number used to identify the specific appointments of each employee’s PHR record. If the employee holds multiple appointments, each record will have a different Internal Id. This is the only key needed to identify the appointment per employee to which this table’s records point. **Joining between Appointment (Appt) tables should be done using Phr_Appt_Internal_Id. You do not need to join on the higher order Institution or U_Id keys.**

- **Curr_Appr_Appt_Ind** – Current Approved Appointment Indicator is a Y/N field indicating whether the employee’s appointment to this position is approved and currently active or not. This indicator is set to Y (Yes) if the current date falls between the Appt_Start_Dt and Appt_Stop_Dt, inclusive, and the Appt_Status_Cd is equal to: A (Active), E (Extended Leave without Pay), or N (Not Paid but Active). This field is set to N (No) if the current date falls outside the Appt_Start_Dt and Appt_Stop_Dt, and the Appt_Status_Cd is equal to: I (Inactive), P (Pending Approval), or R (Reinstated for Payment).

- **Pri_Appt_Ind** – Primary Appointment Indicator indicates whether or not this appointment is the employee’s current primary appointment. An employee may have only one current primary appointment. PHR uses extensive logic when determining this field: 1) if there is only one current appointment, then that is the primary appointment; 2) if there are multiple appointments, then the primary appointment is the PAID appointment which subsidizes the employee’s RETIREMENT; or 3) if there are multiple current, paid appointments which subsidize retirement, the primary appointment is the one with the greatest FTE, or the lowest EEO Code if all the FTE’s are equal; or, 4) if all multiple current appointments are UNPAID, the primary appointment is the one with the greatest FTE, or the lowest Phr_Appt_Internal_Id if all FTE’s are equal.
• **Appt_Status & Appt_Status_Cd**
  
  A – Active
  
  E – Extended Leave
  
  I – Inactive
  
  N – Not Paid, Active
  
  P – Pending
  
  R – Reinstated
  
  S – Sabbatical
  
  W - Terminated, Pending Final Processing

Note – The “W” Appoint Status was added to PHR (11/2005) to accommodate PHR Gross Pay processing. This “W” Appointment status is assigned to terminated appointments after the pay period closes, in which the appointment termination date was effective and approved.

The "W" appointment status keeps the terminated appointment in a **Current** state for a minimum of two full pay periods after the pay period in which the termination was effective and approved. *(Note: A "W" Status appt may remain Current longer than 2 pay periods if there are outstanding approved pay adjustments for that appointment)*. The “W” status was needed in order to process Final Leave Payout payments and also to accommodate changes with an extended time entry closing period. It is important to note that employees with "W" status appointments and no other current/future appts, will continue to display the **Employee Status = "Active"** in the PHR system, as well as in the data warehouse (fieldname: Empl_Status) until the "W" status appointments move to history.

1) The derived field called, **Active_Empl_Ind**, which is only available in the data warehouse, will be = "N" when an employee only has "W" status appointments and no other current or future appointments.
2) Also in the DW when Active_Empl_Ind = "N", the **Separation_Date** will be populated with the latest appointment termination date from the employee's "W" appointment records even though the separation date will not be populated in the PHR system until all final payroll processing is complete.

Therefore, if you are using any of the PHR Employee tables in a query and **ONLY** want to include information on Active employees that have active (not terminated) appointments, you will want to place a limit on **Active_Empl_Ind = 'Y'** in those tables. This will not give you any employees whose current appts have an Appt Status Cd = "W" - Terminating, Pending Final Processing and no other future appts.
• **Pay Method & Pay Method Cd**

  10/22 - 10 Months/22 Pays
  10/12 - 10 Months/12 Months
  95/22 - 95 Months/22 Pays
  95/12 - 95 Months/12 Months
  9/22 - 9 Months/22 Pays
  9/12 - 9 Months/12 Months
  12/12 - 12 Months/12 Months

  E/FPAY - Non-Standard Payment

  HOURLY - Hourly Paid Employee

  NOPAY95 - Non-Paid Employee, 95 Months

  NOPAY12 - Non-Paid Employee, 12 months

  SUM_RSCH - Summer Research

  SUM_PAY - Summer Pay

  SUM_SCHL_1 - Summer School Session 1

  SUM_SCHL_2 - Summer School Session 1a

  SUM_SCHL_3 - Summer School Session 1b

  SUM_SCHL_4 - Summer School Session 2

  SUM_SCHL_5 - Summer School Session 2c

  SUM_SCHL_6 - Summer School Session 2d

  SUM_SCHL_7 - Summer School Session 1/1 Pay

  SUM_SCHL_8 - Summer School Session 2/1 Pay

  OVERLOAD - Teaching Overload, Non-Teaching Overload

  WINTERM - Winter Term
• Position-Driven Data Elements
  o Pos_Num
  o Pos_Cb_Code
  o Category_Status
  o Category_Status_Cd
  o Title
  o Title_Cd
  o Unit
  o Unit_Cd
  o Eeo
  o Eeo_Cd

• Appointment Salary/FTE
  o Appt_Base_Annual_Salary
  o Appt_Base_Bwkly_Salary
  o Hour_Rate
  o Appt_Fte

• Appointment Status/Dates
  o Appt_Status
  o Appt_Status_Cd
  o Appt_Start_Dt
  o Appt_Termin_Dt

• Extended Leave Info
  o Extnd_Lv_Type
  o Extnd_Lv_Type_Cd
  o Extnd_Lv_Start_Dt
  o Extnd_Lv_End_Dt

• Faculty Leave Info
  o Curr_Fac_Leave_Flag
  o Fac_Leave
  o Fac_Leave_Pct_Pay
  o Fac_Leave_Start_Date
  o Fac_Leave_End_Date

* Note: This table does not yet retain data on employees after the Pay Period in which they separated from the University has passed. We are in the process of adding this (and future-related) information, and will let you know when it is available for your use.
Exercise 2: Review Phr_Dv_Appt Table

1) Insert a new query (Query 2).

2) From the Section/Catalog, **Open the Phr_Dv_Appt table**.

3) **Drag** the Appt Data Elements to the Request Line.

4) **Limit** on your Unit_Cd.

5) **Click on Process** and **Review** your results.

Be prepared to deal with the possibility of multiple current appointments per individual. You may want to add an additional limit on **Pri_Appt_Ind = Y** (that’s Primary Appointment Indicator = Y) if you want to restrict your Results to the one Primary appointment of the queried population. If you want to retrieve **all** of the appointments which an employee currently holds, then eliminate the Pri_Appt_Ind limit.
PHR_EDV is a secured data mart in the PHR Data Warehouse. This table pulls information from other data warehouse tables including the Employee Demographic (Phr_Empl_Demog), Appointment and Appointment Funding tables. This is a very useful table to obtain a complete detailed picture of all current active employees and appointments.

Exercise 3: Review Phr_EDV Table

1) **Insert** a new query (Query 3).

2) From the Section/Catalog, **Open** the Phr_EDV table.

3) **Drag** the EDV Data Elements to the Request Line.

4) **Limit** on your Unit_Cd.

5) **Click on Process** and **Review** your results.
Limits

When building your own queries, always limit on the **CODE** not the DESC – this avoids query errors due to mistyped limits. If you are uncertain as to what the exact code should be for your limit, use the Show Values functionality to view the list of all the possible coded values and matching short descriptions for the data element and then select the appropriate choices.

Common PHR Limits

**Appt_Status_Cd = A (Active)**

Appt Status Cd is a PHR (Payroll Human Resources) System code which indicates the current appointment's activity status.

**Empl_Status_Cd = A (Active)**

Empl Status Cd identifies the current status of the employee at the time of the load of PHR employee data to the DW.

**Curr_Appr_Appt_Ind = Y (Yes)**

Curr Appr Appt Ind is a Y (Yes) / N (No) field which indicates if the employee’s appointment to this position is approved and currently active or not. This indicator is set to Y (Yes) if the current date falls between the Phr_Appt_Start_Dt and Phr_Appt_Stop_Dt, inclusive, and the Appt_Stat_Cd is equal to: Active (A), Extended Leave without Pay (E), or Not Paid but Active (N). This indicator is set to N (No) if the current date falls outside of the Phr_Appt_Start_Dt and Phr_Appt_Stop_Dt and the Appt_Stat_Cd is equal to: Inactive (I), Pending Approval (P), or Reinstated for Payment (R).

**Pri_Appt_Ind = Y (Yes)**

Pri Appt Ind indicates whether or not this appointment is the employee's current primary appointment. An employee may have only one current primary appointment.

**USM_Inst_Cd = 01**

USM Inst Cd is a two digit code which represents the University System of Maryland Institutions. For PHR, this includes only those institutions which also use the PHR software. They include UMCP (01), UMBI (03), UMES (04), UMCES (07), and USMO (08).

**Pay Method Cd != (not equal to) NOPAY95, NOPAY12 and E/FPAY**

A common limit when providing current employee data is to limit on Payment Method to exclude No Pay and Non-Standard Pay employees from your population.
Joins

Joins are a central concept of the relational database. Joins enable you to connect or link records in two tables by way of a shared data field. Once a data field is shared, other data contained in the joined tables can be accessed. In this way, each record can share data with another record, but does not store and duplicate the same kind of information.

Types of Joins

1. **Simple Join** - A simple (linear) join retrieves the records in both tables that have identical data in the joined columns.

2. **Left Join** - A left join captures all records from your driver table and any record(s) (if present) from the joined table. If no record is present in the joined table, the employee’s record (row) in Results will appear blank for all data elements pulled from the joined table.

3. **Right Join** - A right join retrieves all rows from the table on the "right" and any rows from the table on the "left" which have matching values in the joined column.

4. **Outer Join** - An outer join retrieves all rows from both tables matching joined column values if found or retrieves nulls for non-matching values.

Joining between any Employee tables (including PHR_DV_Employee) and/or between any Employee table to any Appointment table (including PHR_DV_Appt) should always be done by joining USM_Inst_Cd and U_Id. Because U_ids in employee's PHR records will all be under the same U_Id, regardless of which institutions he or she has worked, you must join on the Institution (Usm_Inst_Cd) as well as the U_Id. If you want to limit to just your home institution, apply the limit to the table on the left side of your query's data model, then join on Usm_Inst_Cd from that table to any others **(You do not have to place the institution limit on the other tables, just on the main driver table.)** Your data warehouse security should protect you from mixing up appointments between institutions, but it is still probably a good habit to join on USM_Inst_Cd anyway.

Joining between any Appointment tables (including PHR_DV_Appt) to any other Appointment table (i.e. PHR_Appts_Funding_Accts) should be done using Phr_Appt_Internal_Id. You do not need to join on the higher order Institution or U_Id keys because Phr_Appt_Internal_Id is unique to each appointment. However, you can also join Appt tables on USM_Inst_Cd and U_Id if you want.
Note: *Phr_Appt_Internal_Id* is the data warehouse name for 'Internal ID', which is the Id that you see on the PHR appointment selection list and all of the other PHR appointment screens in the production system. It appears in the upper left-hand corner of the PHR Appointment screen.

**Exercise 4: Retrieve All Current Active Employees and Total Salaries (Using Phr_Edv from 3rd Query)**

Goal: To become familiar with using simple and left joins.

Tables Used: Phr_EDV, Phr_Empl_Total_Salary

Request: U Id (from Phr_Edv)

 Primary_Name (from Phr_Edv)
 Category_Status (from Phr_Edv)
 Pay_Method_Cd (from Phr_Edv)
 Total_FTE (from Phr_Edv)
 Total_Base_Annual_Salary (from Phr_Edv)
 Total_Rpt_Annual_Salary (from Phr_Empl_Total_Salary)
 Total_Salary12 (from Phr_Empl_Total_Salary)
 Total_Salary9 (from Phr_Empl_Total_Salary)

Limits: Usm_Inst_Cd = 01 (from Phr_Edv)

 Curr_Appr_Appt_Ind = Y (from Phr_Edv)
 Pri_Appt_Ind = Y (from Phr_Edv)

Joins: USM_Inst_Cd

U Id
1. From the Section/Catalog, **Open the Phr_Empl_Total_Salary** table.

2. **Join** (simple ‘=’) your tables by USM_Inst_Cd and UID.

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

*Take note of the number records - Because this is an equal join to the Total Salary table – you will not pull in any records for hourly or non-paid employees. The Total Salary table does not include records for these employees.*

Joins: Using a left join ‘+=’

Return to the Query Section and Change the Join Type to a Left Join as follows:

1. **Double click** on each join line.

2. **Select the left join option** button from the Join Properties Pop Up Box.
3. Click OK.
Once you have changed both joins, your query should look like this:

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

Notice that you may have more rows returned when using left joins if there is more information in your driver table.

While in the Results Section;

5. **Drag the Pay_Method_Cd** field to the limit line

6. **Select “Hourly”** from the available values, then **Click OK**.
7. **Review** the results.

*With a left join between these two tables, you are now including Hourly employees. Hourly employees are included in the Phr_Edv table, but not in the Phr_Empl_Total_Salary table, so the left join included these hourly employees in your results.*
Common Computed Items

**Dup Checker** - Used to remove duplicate rows of data.

**Exercise 5**: Removing duplicate appointment records.

*(Using DV_Employee and DV_Appointment tables from the 1st query)*

Tables Used: Phr_DV_Employee, Phr_DV_Appt

Request: Primary_Name (from Phr_DV_Employee)

- U Id (from Phr_DV_Employee)
- PHR_Appt_Internal_ID (from Phr_DV_Appt)
- Pri_Appt_Ind (from Phr_DV_Appt)
- Unit_Cd (from Phr_DV_Appt)
- Unit (from Phr_DV_Appt)
- Title (from Phr_DV_Appt)
- Category Status (from Phr_DV_Appt)
- Appt FTE (from Phr_DV_Appt)
- Pay_Method_Cd (from Phr_DV_Appt)

Limits: Usm_Inst_Cd = 01

- Curr_Appr_Appt_Ind = Y

Joins: Using a simple join ‘=’, join the following fields:
1. **Remove all items from the Request and Limit Lines** if there are any there.
2. **Add** the above data fields to the Request and Limit lines.
3. **Join** your tables by USM_Inst_Cd and UID.

Your query should now display as:

![Database query interface](image)

4. **Process** the query.
5. **Sort** your results by **U_Id** – **this is very important!**
   (If data is not sorted by the field in which you are performing the dup checker function, the end results from this function will be incorrect.)
6. **Highlight the U_Id column and Right-click,**

7. **Select the Add Computed Item** from the menu.
8. In the Computed Item window, **Place** your cursor in the definition box.

9. **Enter** the following statement in the definition box:

   ```
   if (U_Id == Prior ( U_Id )) {'dup'} else {'ok'}
   ```

10. **Change** the column name to an appropriate name (i.e. Dup Checker)

11. **Click OK** when your computed information is correct.

    *When using the **Prior** function, any limits on this column must be performed in a new Table; therefore,*

12. **Insert New Table** by Clicking on Insert/New Table from the main menu bar.

13. **Drag** all fields from Results into the new Table.

14. **Limit on the Dup Checker** computed item - Dup Checker != (not equal) dup.
15. **Review** your results. Each employee should now have one record.

**Grouping Column** – A grouping column is used to pull records together by new defined groups. (i.e. certain dates, all faculty Cat Stats etc.)

**Exercise 6: Current Employee vs. Appointment Counts**

Goal: Learn to create a Grouping Column. Obtain counts of current active employees and current active appointments by employment grouping

Table Used: Phr_Dv_Appt

Request: U_Id

<table>
<thead>
<tr>
<th>Primary_Name</th>
<th>Unit</th>
<th>Category_Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Category_Status_Cd

Phr_Internal_Id

Limits:  Usm_Inst_Cd = 01
           Curr_Appr_Appt_Ind = Y

1. Insert a new query.

2. Drag the above items to Request Line.

3. Add the above limits to the Limit Line.

Your query should display as:
4. **Process** the query and review the results.

*For this exercise we will group the employees by Category Status.*

5. **Highlight** the Category Status column and then **Right-click** your mouse.

6. Select **Add Grouping Column** from the menu.
7. **Add** each Group by **Clicking** on the New Group button.

Once you have entered your groups:

8. **Click** on the first group.

9. **Highlight** the Available items appropriate for that group.

10. **Click on the <<** button to move the items into the Items in Group box.
11. **Drag** the appropriate values to each group as appropriate.

12. **Click on OK** when you are completed.

13. **Review** the new column – **Group of Category Status**.

   *(Notice how each employee is grouped into a separate category, based upon Category Status)*

To Review the actual statistics of number of employees versus appointments, you will want to use a **Pivot Report**.

14. **Insert a Pivot** by selecting Insert/New Pivot from the main menu.

15. **Drag** the Group of Category Status and Category Status fields into the **Side Labels** of the pivot outliner.

17. **Drag** the U Id and PHR Appt Internal ID into the **Facts** section of the outliner.
18. **Right Click** on the U Id column and PHR Appt Internal ID in the Facts and Choose the **Data Function of Count Distinct**.

You will then have counts of employees and appointments by their employment group.

19. **Review** the results with the new column.

(Notice how the number of employees within each group can be less than the number of appointments. This is because one column gives you a distinct employee count and the other column gives you a distinct appointment count. Many employees have multiple appointments, therefore the appointment count is usually higher.)
You can **Add Grouping Totals** by:

20. **Click** on the top bar of the second column and then **Right-click** your mouse to view the pivot menu (or you can **Click** on Pivot on the main menu bar).

21. **Select Add Totals**.

22. **Review** the grouping totals results – compare employee vs appointment totals by Category Grouping.
**Tips & Tricks**

➤ **Auto-Join** – this feature should be turned off to avoid automatic & unanticipated table joins when opening more than one table on your desktop.

1. Select **DataModel** menu.

2. Make sure that **Auto-Join** (in pull down menu) is *not* checked (toggled off).
   
a. If it is checked, click on Auto-Join and this will remove the check mark.

➤ **Blank fields** in the Transactional Systems are noted in the Data Warehouse according to the following:

- Alpha fields (text fields) = * (asterisk)

- Code fields = * (asterisk)

- Numeric fields = (null)

- Date fields = (null)

➤ Remember when limiting on **Yes/No** fields that they are always either Y or N.

➤ **Show Values** feature lists all the possible coded values and matching short descriptions for the data element. Make sure you only use this feature when appropriate, i.e. Category Status Cd, for which there are a limited number of values, *not* Appt Salary, for which there exists an enormous number of unique values.

➤ **Remarks** are present in the Data Warehouse to give users a better understanding of a particular field within a table:

1. Open a table.
2. Highlight the field name and
   a. Press <CTRL><I> to view “remarks” OR
   b. Right-click on the field and choose Show Remarks.

➢ To alphabetize the data elements in a table on the workspace:

1. Open a table.
2. Double-click on the table name.
3. Press the Sort button.
4. Click on OK.

➢ Terminated Appointments are stored in the current Data Warehouse for the current pay period only.

➢ Primary Appt Indicator (Pri_Appt_Ind) vs Pay Period Primary Appt Indicator (PP_Pri_Appt_Ind):

   Pri_Appt_Ind indicates whether an employee’s appointment is the current primary appointment. (This is based upon many appointment factors (Appt_Fte, etc.)

   PP_Pri_Appt_Ind indicates whether an employee’s appointment is the primary appointment for the current pay period.

An employee may have only one current primary appointment (Pri_Appt_Ind) but he/she may have another appointment that is terminating during the Pay Period in question that was primary when it was still active. PP_Pri_Appt_Ind is used for payroll purposes and should not be used by routine queries that are looking for the Primary Appointment; Pri_Appt_Ind is the preferred field. PHR uses extensive logic when determining this field; the DW extract program draws Pri Appt Ind directly from PHR_POSITION.Pri_Pos_Ind.

➢ One to Many Relationships – the Need for a Left Join:

Remember that an employee may have one record in one table and many records (or no records) in your joining table.
Example: Your driver table is Phr_Appointments and is joined to Phr_Appt_Sal_Additions. Employee A has one record (one appointment) in Phr_Appointments and three salary addition records in Phr_Appt_Sal_Additions. Employee B has one record in Phr_Appointments and NO records in Phr_Appt_Sal_Additions (no salary additions for appointment). In order to capture ALL information for both employee’s records in both tables, you need to change the join between the two tables to a left join.

1. Double-click on the join between the two tables.
2. Click on **Left**.
3. Click **OK**.

A left join captures all records from your driver table and any record(s) (if present) from the joined table. (If no record is present in the joined table, the employee’s record (row) in Results will appear blank for all data elements pulled from the joined table.

**Reference**

<table>
<thead>
<tr>
<th>Employee Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Active</td>
</tr>
<tr>
<td>E</td>
<td>Only Emeritus Status</td>
</tr>
<tr>
<td>F</td>
<td>Future Appointment</td>
</tr>
<tr>
<td>I</td>
<td>Inactive</td>
</tr>
<tr>
<td>N</td>
<td>Returning to Academic Appointment</td>
</tr>
<tr>
<td>Q</td>
<td>Only Non-Paid Appointment</td>
</tr>
<tr>
<td>R</td>
<td>Reinstated for Payment</td>
</tr>
<tr>
<td>T</td>
<td>Terminated with a Future Appointment</td>
</tr>
<tr>
<td>X</td>
<td>Incomplete</td>
</tr>
</tbody>
</table>

**Appointment Status**

<table>
<thead>
<tr>
<th>A</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Extended Leave Without Pay</td>
</tr>
<tr>
<td>I</td>
<td>Inactive, terminated record</td>
</tr>
<tr>
<td>N</td>
<td>Not Paid but Active, Returning, i.e. 10 month employee</td>
</tr>
<tr>
<td>P</td>
<td>Pending Approval</td>
</tr>
<tr>
<td>R</td>
<td>Reinstated for Payment</td>
</tr>
</tbody>
</table>
**Category Status Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Faculty Tenured</td>
</tr>
<tr>
<td>02</td>
<td>Faculty Non-Tenured, on Track</td>
</tr>
<tr>
<td>03</td>
<td>Faculty Non-Tenured, Term Contract</td>
</tr>
<tr>
<td>04</td>
<td>Graduate Assistant, on a Bi-Weekly Salary</td>
</tr>
<tr>
<td>05</td>
<td>Fellow</td>
</tr>
<tr>
<td>14</td>
<td>Student</td>
</tr>
<tr>
<td>15</td>
<td>Faculty Non-Tenured, Continuing Contract</td>
</tr>
<tr>
<td>16</td>
<td>Graduate Student Hourly</td>
</tr>
<tr>
<td>19</td>
<td>Trainee Employee</td>
</tr>
<tr>
<td>20</td>
<td>Nonexempt, Regular</td>
</tr>
<tr>
<td>22</td>
<td>Nonexempt, Contingent Category 2</td>
</tr>
<tr>
<td>25</td>
<td>Faculty, Contractual</td>
</tr>
<tr>
<td>31</td>
<td>Nonexempt, Contingent Category 1</td>
</tr>
<tr>
<td>33</td>
<td>Exempt, Regular</td>
</tr>
<tr>
<td>34</td>
<td>Exempt Contingent Category 1</td>
</tr>
<tr>
<td>35</td>
<td>Exempt Contingent Category 2</td>
</tr>
<tr>
<td>36</td>
<td>Faculty, Hourly</td>
</tr>
<tr>
<td>37</td>
<td>Faculty, Non-tenured, Non-regular</td>
</tr>
</tbody>
</table>
### Appointment Actions (Refer to handout for definition details)

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/REG</td>
<td>Cont to Regular</td>
</tr>
<tr>
<td>CHANG</td>
<td>Change to Appointment</td>
</tr>
<tr>
<td>CTOC</td>
<td>Cont to Cont</td>
</tr>
<tr>
<td>DEMOT</td>
<td>Demotion</td>
</tr>
<tr>
<td>INEMP</td>
<td>Initial Employment</td>
</tr>
<tr>
<td>LATRL</td>
<td>Lateral</td>
</tr>
<tr>
<td>PROMO</td>
<td>Promotion</td>
</tr>
<tr>
<td>PROMW</td>
<td>Promotion W/out Search</td>
</tr>
<tr>
<td>R/CONT</td>
<td>Regular to Cont</td>
</tr>
<tr>
<td>REAPPT</td>
<td>Reappointment</td>
</tr>
<tr>
<td>REASG</td>
<td>Reassignment</td>
</tr>
<tr>
<td>RECLSD</td>
<td>Reclassification-Demotion</td>
</tr>
<tr>
<td>RECLSP</td>
<td>Reclassification-Promo</td>
</tr>
<tr>
<td>REEMP</td>
<td>Re-Employment</td>
</tr>
<tr>
<td>REINS</td>
<td>Re-Instatement</td>
</tr>
<tr>
<td>TTLCH</td>
<td>Title Change</td>
</tr>
<tr>
<td>XNEWAP</td>
<td>New Appointment</td>
</tr>
</tbody>
</table>
Additional Practice Exercises

Additional Exercise 1 - Retrieve All Business Addresses for Employees

Goal: To become familiar with the Dv_Appt and Dv_Employee tables by running a query on employee’s business addresses. To become acquainted with joins and limits within the query and results sections.

Tables Used: Phr_Dv_Appt, Phr_Dv_Employee

Request: Name_First
          Name_Last
          Unit
          Room
          Ups_Bldg
          Umcp_Zip
          Email_Address

Limits:  Usm_Inst_Cd = 01
          Curr_Appr_Appt_Ind = Y
          Pri_Appt_Ind = Y
1. **Drag** the listed request and limit fields above to the Request and Limit lines.

2. **Join** the Usm Inst Cd and UID fields using an = join.

3. **Click on Process** and **Review** the results.
Additional Exercise 2 - Using the *Timefunc* computed item.

*Timefunc* – Used to display HH:MM instead of the number of minutes. This should be done in the query (from the request line) before processing. Be sure to set the Datatype to String – click Options to see Datatype.

Goal: To learn the technique of using the *timefunc* function to convert time fields (stored in the warehouse in minutes) to HH:MM.

Tables Used: Phr_Dv_Appt and Phr_Dv_Employee

Request: U_Id
          Primary_Name
          Leave_Bank_Fiscal_Yr
          Leave_Bank_Pay_Period
          Leave_Bank_Annl_Mnt
          Leave_Bank_Sick_Mnt
          Leave_Bank_Pers_Mnt

Limit:   Usm_Inst_Cd = 01
          Curr_Appr_Appt_Ind = Y
          Category_Status_Cd = 20, 33

Computed Items: *Timefunc*
To perform the timefunc function:

1. **Drag** the listed request and limit fields above to the Request and Limit lines.

2. **Double-click on the Leave_Bank_Annl_Mnt** field from the Request line.

3. In the Item Properties Definition box, **Type timefunc** before the Phr table/field name. *Be sure to put the table/field name in parenthesis.*

4. **Click on Options** to change the Datatype to String.
5. **Process** the query.

6. **Review** the results.

Notice how the Leave_Bank_Annl_Mnt field is now displaying in HH:MM, as opposed to only minutes like the other Leave_Bank fields.
Additional Exercise 3 – Exporting/Importing to and from Excel

Goal: To learn the functionality of exporting and importing data to and from Brio and then joining the information to an existing Data Warehouse Table.

Exercise – Export a list of 5 employees to an Excel spreadsheet, add vacation request information and then import that Excel document back into Brio. Join those results with the PHR Data Warehouse employee table and find out their Continuous Employment Date so you can determine who has seniority for vacation time.

1. **Open** a new blank Brio Query by selecting File/New
2. **Insert a Query** by Clicking on Insert/New Query from the menu bar.
3. **Open your Tables** in the Section Catalog, and add the PHR_DV_APPT table.
4. **Drag** the following items to the Request Line: **U_Id and Primary_Name**
   * Curr_Appe_Appt_Ind = “Y”
   * Unit_Cd = “Select one of your unit codes”
   * Pri_Appt_Ind = ‘Y”
5. **Limit the number of rows returned** by Selecting Query/Query Options and then check the Box in front of Return First Rows. **Enter** the number “5” for the number of rows. **Click OK** when done.
7. **Process the Query.** You will receive a message that the “maximum number of rows retrieved”.
8. **Click OK.** There should be 5 rows of data displayed in your results.

9. From the Results Section, Select **File/Export/Section.**
10. The Export Section Pop Up will display, **Select the folder** location to export the file, **Enter a Name** for your spreadsheet results i.e. “Exercise 3 Export” and then **Click on Save**.
11. **Open** your exported results using the Excel application.

12. **Add an Additional Column** of information to your Excel document labeled “Vacation Requests” and add Vacation Dates next to each name.

13. **Save your document** by selecting File/Save. You will receive a pop up that asks you if want to save the document in the latest Excel format. Click “Yes” and close your Excel document.

**Important Step:** When results are exported from Brio to Excel, the information exports in the lowest version of Excel - version 2.1. If you need to re-import the excel document back into Brio, you **must** save your document in the highest Excel version (usually Microsoft Excel 97-2002).

14. **Click** back to your Brio Query.

15. **Import** your updated Excel spreadsheet into Brio by selecting File/Import Data File/Data File from the main menu.
17. **Locate** your Excel document to import. (Be sure that the File Type at the bottom is selected to Excel File.)

18. **Click Open** after you select your document. Your spreadsheet will appear as a new Results section.
Join your new results to the PHR_Dv_Employee Table.

19. **Insert** a new query section by selecting **Insert/New Query** from the menu bar.

20. **Login** to the query using the DW connection.
21. **Right Click** on the “Tables” in the Section Catalog. A Local Results pop up should appear. Click on the Local Results pop up, and open the Local Results tables.

22. Drag your Excel Results into the blank query section.
23. **Open your “Tables”** by clicking on the “Tables” in the Section Catalog.

24. **Locate the PHR_DV_Employee** table and Drag that table next to the Excel Results.

25. **Join** the tables by U_ID.
Drag to the Request Line: U_ID, Primary Name and Vacation Request from the Exercise 3 export table; and Cont_Employment_Date from the PHR_Dv_Employee table. Click on Process.
27. **Sort** Results by Continuous_Employment_Date. Now you can determine who to grant vacation time to based on Continuous Employment Date.