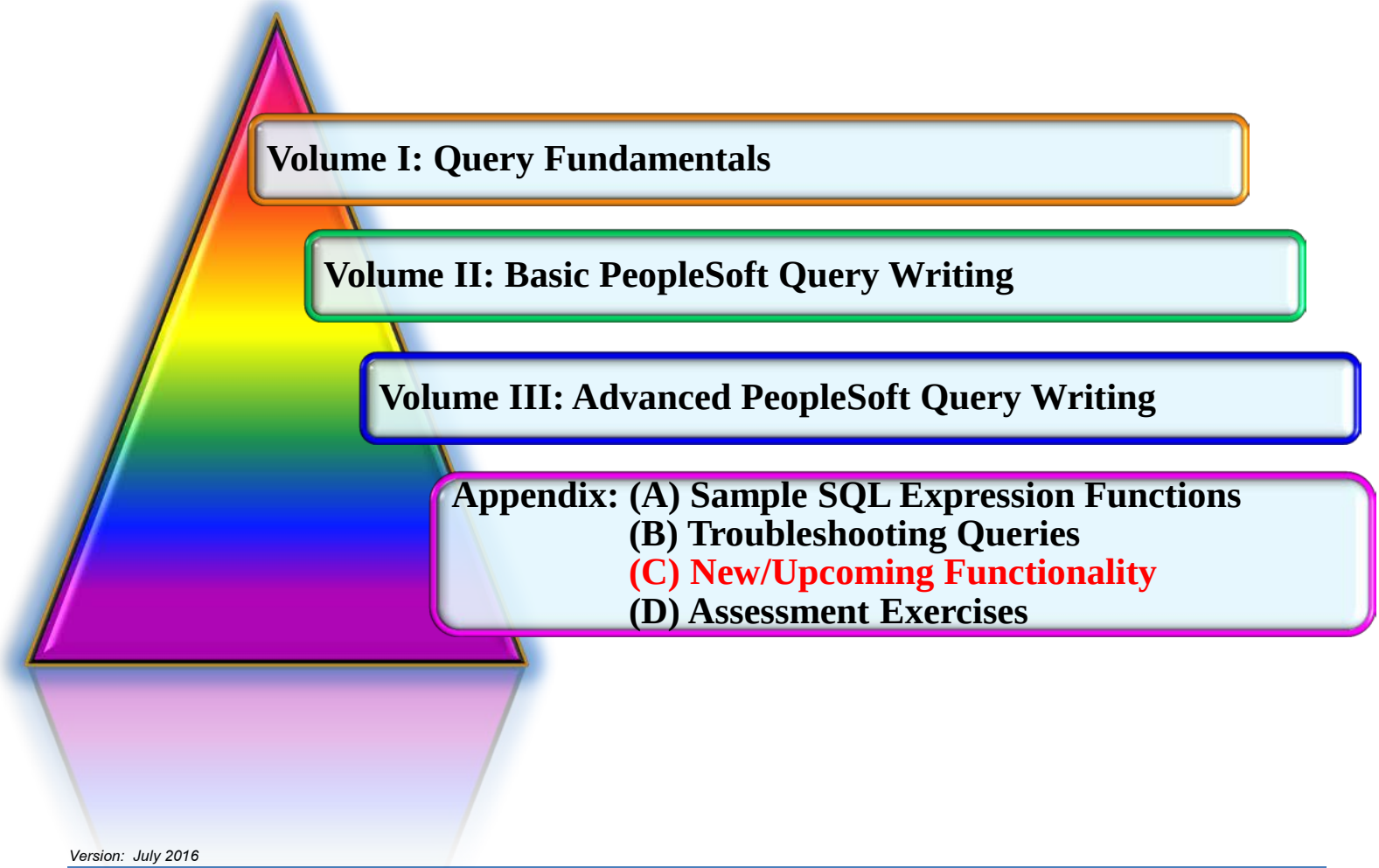
UNIVERSITY of HOUSTON SYSTEM

PeopleSoft Query Training Guide



Version: July 2016

Introduction

Agenda

- Introductions
- Training Objectives and Material Overview
- University of Houston System Environment
    - PeopleSoft Databases
    - Reporting Tables
    - CB Tables
- Query Fundamentals
- Query Methodology
- PeopleSoft Query Structural Process Flow
- Writing Basic PeopleSoft Queries
- Writing Advanced PeopleSoft Queries
- Query Workshop

Objectives

Welcome to Basic PeopleSoft Query Training! This class contains the resource information needed to learn the concepts and procedures related to the PeopleSoft Query toolset. The PS Query toolset provides the easiest mechanism to create and generate simple reports from PeopleSoft. While PS Queries cannot be used to produce highly formatted and graphical reports, it is very effective in extracting day-to-day inquiry queries that are of interest to employees, managers and administrators. At the end of this class, participants will:

- Understand the UHS environment and reporting tables
- Understand the PeopleSoft Query tool and how to navigate it
    - Query Viewer vs Query Manager
    - Output Formats
    - Global Search Features
    - Searching for Queries
    - Running Queries
    - Downloading Queries
    - Scheduling Queries
    - Favorite Queries
    - Report Manager
- Understand general Query Methodology
- Understand PeopleSoft Query Structural Process Flow
- Query Manager Design Pages and Page Links
- Understand how to develop simple queries
    - Selecting data from a single table
    - Specify the column order
    - Specify the sort order
    - Specify criteria for retrieving data
- Understand how to incorporate advanced query features
    - Aggregate Functions
    - Having Criteria
    - Prompts
    - Expressions
    - Join Multiple Tables
    - Subqueries
    - Unions

Overview

This class will first introduce users to the University of Houston System reporting environment. During this introduction we will discuss PeopleSoft databases, custom reporting and CB tables. The PeopleSoft Query tool will then be introduced and users will be shown how to pull up and run queries.

Basic query functionality will then be demonstrated and users will develop and run a basic query. Users will learn to develop simple queries by selecting from a single record, changing field headings, adding criteria, and sorting the records.

More advanced query concepts and procedures will then be introduced to users. Users will learn to incorporate more advanced query functionality by working with aggregate functions, having criteria, utilizing prompts, creating expressions, adding record joins, and utilizing subqueries and unions.

Table of Contents

(Content Below Is Hyperlinked Within the Guide)

VOLUME I: Query Fundamentals 8

CHAPTER 1 – UHS Databases and Reporting Tables 9

SA9RPT – Reporting Database 10

Reporting Tables 11

Reporting Tables – Data Dictionaries 11

Housing Tables – Data Dictionary 41

CB Tables – Data Dictionaries 45

CHAPTER 2 – Logging In 52

CHAPTER 3 – PeopleSoft Query Elementals 53

Query Viewer vs Query Manager 53

Query Reporting Formats 53

PeopleSoft Global Search Features 53

Types of Searches 53

Using Operators 54

Using Wildcard Characters 55

Searching for Queries 55

'Query Manager' Search Page Actions 58

Using Effective Data in Queries 59

Running Existing Queries 60

Running an Existing Query with a Prompt 61

Downloading Queries 63

Scheduling Queries 64

'Process Scheduler Request' Page Objects: 66

Report Manager 69

My Favorite Queries List 70

VOLUME II: Basic PeopleSoft Query Writing 72

CHAPTER 4 – Query Writing Essentials 73

Query Methodology 73

PeopleSoft Query Structural Process Flow 73

CHAPTER 5 – Query Manager Design Pages and Page Links 74

Summary Page Descriptions 74
'Records' Page 76
'Query' Page 77
'Expressions' Page 79
Using Drilling URLs (Special Type of Expression) 81
Using Dates in Expressions 82
'Prompts' Page Tab 84
'Fields' Page Tab 87
'Edit Field Ordering' Subpage Objects 88
'Edit Field Properties' Subpage Objects 88
'Criteria' Page Tab 90
'Having' Page Tab 91
'View SQL' Page 92
'Run' Page 92
'Query Preferences' Page 93
'Query Properties' Page 96
'Save/Save As' Page 98
CHAPTER 6 – Basic Query Writing 100

VOLUME III: Advanced PeopleSoft Query Writing 113
CHAPTER 7 – Creating Queries with Aggregate 114
CHAPTER 8 – Creating Query with Having Criteria 118
CHAPTER 9 – Creating Queries with Prompts 122
CHAPTER 10 – Creating Expressions 131
CHAPTER 11 – Joining Multiple Tables 139
Understanding Join Categories and Join Types 139
Record Hierarchy Join 139
Related Record Join 140
Any Record Join *(and Cartesian Joins)* 141
Inner Join 142
Outer Join 142
Creating a Query with Multiple Records 142
Creating an Outer Join 148
CHAPTER 12 – Creating Queries that Use Subqueries 154
Subquery Basics 154
Subquery Rules 154

Creating a Subquery Query 154
TIP: Subquery Navigation 158
CHAPTER 13 – Creating Unions 160
Union Queries Basics 160
UNION (Distinct) versus UNION ALL 160
Union Query Rules 160
Creating a Union Query 161
TIP: Union Query Navigation 169

APPENDIX: Sample SQL Expression Functions 170

String Functions 171
CONCAT(str1, str2) ; str1 || str2 171
INITCAP(str1) 171
INSTR(str1,str2,[start position],[nth_appearance]) 171
LENGTH(str1) 172
LOWER(str1) 172
REPLACE(str1,str_to_replace,[replacement_str]) 172
SUBSTR(str1, start[, how_many]) 172
UPPER(str1) 172
Aggregate Functions 173
Numeric Functions 173
ABS(n) 173
CEIL(n) 173
FLOOR(n) 173
ROUND(n[, dec]) 173
SQRT(n) 174
TRUNC(n[, dec]) 174
MOD(m,n) 174
REMAINDER(m,n) 174
ROWNUM 174
Date Functions 175
ADD_MONTHS(dt, m) 175
SYSDATE 175
MONTHS_BETWEEN(date1, date2) 175
NEXT_DAY(date, char) 175
TRUNC(datetime) 175

Conversion Functions 176
TO_CHAR(value[, fmt]) 176
Condition Functions 178
DECODE(x, val1, [val2, result2, …, valN, result] [, default]) 178
COALESCE(expr1, expr2, …, expr_n) 178
NVL(string1, replace_with) and NVL2(string1, value_if_not_null, value_if_null) 179
CASE Expression 179
GREATEST(expr[,expr]) 181
LEAST(expr[,expr]) 181
Analytic Functions 182

APPENDIX: Troubleshooting Queries 182
If the query produces many more rows than are expected **184**
If the query produces no results or many fewer rows than are expected **184**
Problems with expressions **184**
Problems with the Query Hanging, Looping, Returning Infinite Rows (Cartesian Joins) **185**
Problems with Unions **185**
Problems with Prompts **185**
Problems with Aggregates **185**
Unable to find a query **186**
Look at the SQL tab! **186**
Query doesn't Return Results to Excel (using IE Browser) **1866**

APPENDIX: New/Upcoming Features to be Incorporated Soon 187
APPENDIX: Assessment Exercises 189

VOLUME I:

Query Fundamentals

Objectives

Participants will learn the following concepts and procedures:

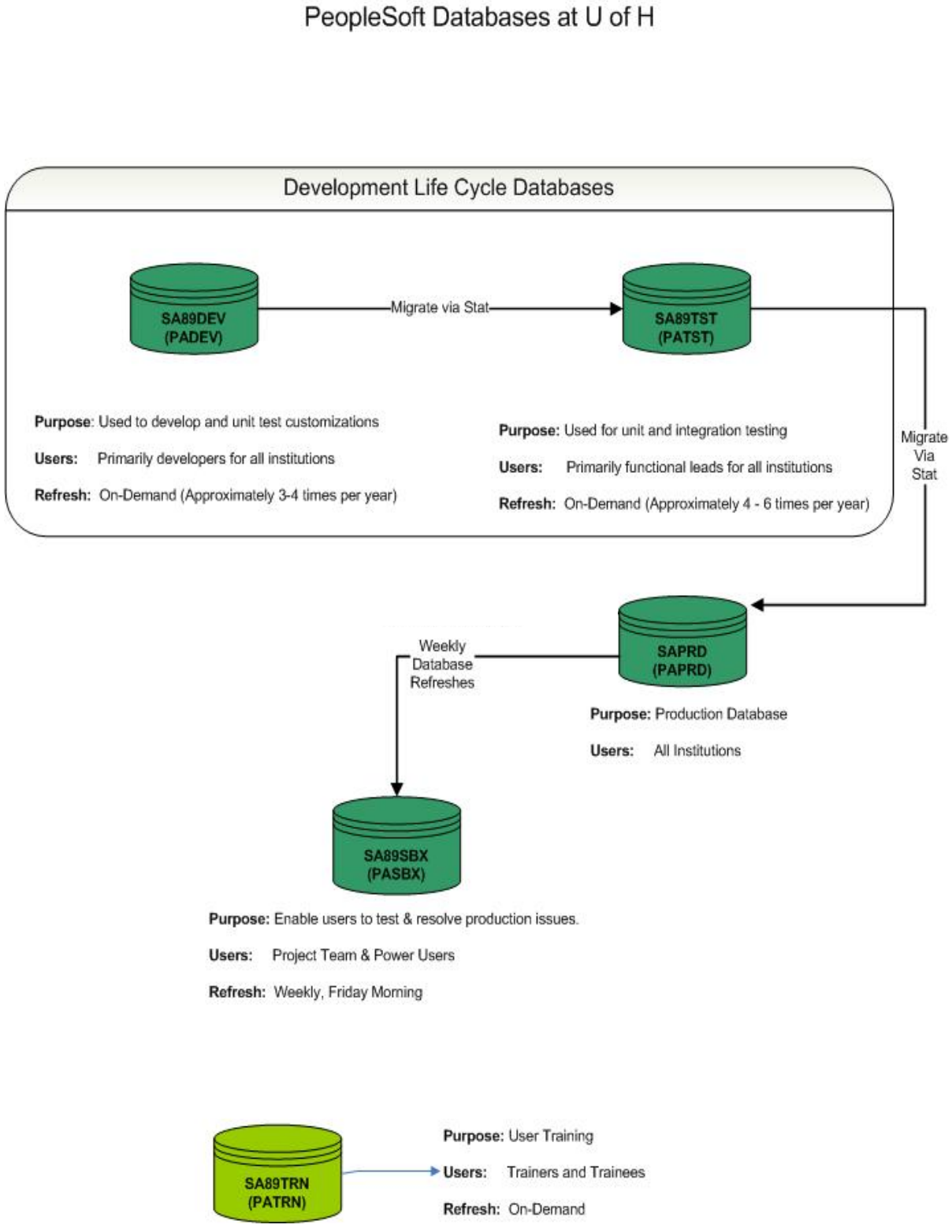
- Understand the UHS environment and reporting tables
- Understand the PeopleSoft Query tool and how to navigate it
    - Query Viewer vs Query Manager
    - Output Formats
    - Global Search Features
    - Searching for Queries
    - Running Queries
    - Downloading Queries
    - Scheduling Queries
    - Favorite Queries
    - Report Manager

Overview

Users will be introduced to the University of Houston System reporting environment. During this introduction we will discuss PeopleSoft databases, custom reporting tables and CB tables.

The PeopleSoft Query tool will then be introduced and users will be shown fundamental functionality offered by the tool.

CHAPTER 1 – UHS Databases and Reporting Tables



SA9RPT – Reporting Database

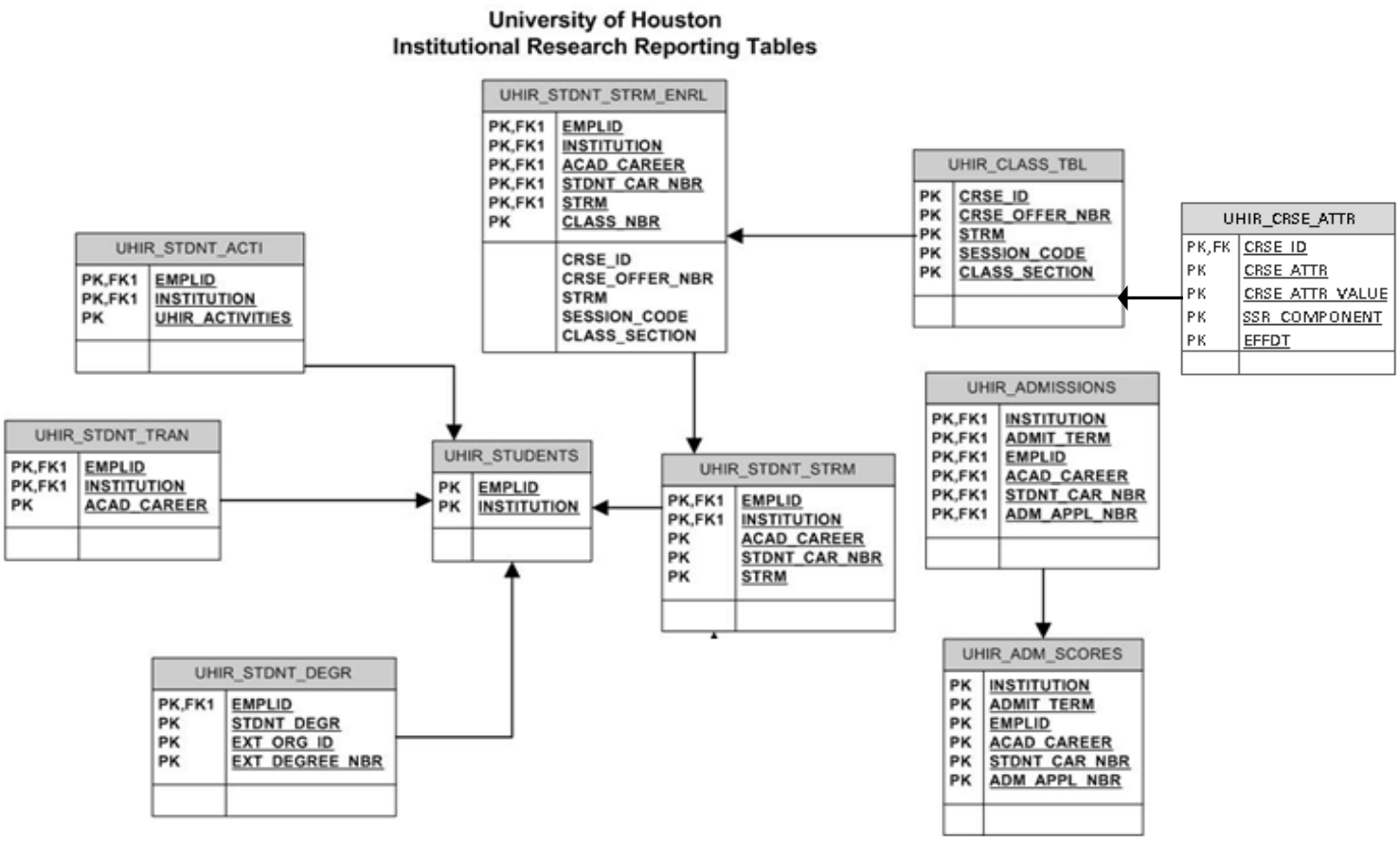
Database Purpose:
As of December 2013, the SA9RPT database is **no longer used** by Student Administration users for query writing and reporting purposes.

Reporting Tables

Purpose: The UHIR Reporting Tables facilitate the development of queries by consolidating data into less than a dozen reporting tables. The data in these tables can be used to report as of census, current or end of semester enrollments.

Key Concepts:

- **Denormalized Data**: Data that has been mass consolidated. A fully normalized database schema can fail to provide adequate system query response time and add substantial query complexity due to excessive table join operations whereas denormalization can speed data retrieval. Denormalization must balance the need for good system response time with the need to organize data, while avoiding the various anomalies or problems associated with denormalized table structures such as excessive redundancy.
- **Normalized Data**: Data that has undergone the process of organizing the data to minimize redundancy. Normalization usually involves dividing a database into various tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships. The relationships make it possible to store information in fewer locations within the database.
- **Relational Database**: A database that maintains a set of separate, related files (tables), but combines data elements from the files (tables) for queries, views, and reports when required.
- **Primary Key**: The primary key of a relational table uniquely identifies each row in the table. It can either be a normal attribute that is guaranteed to be unique (such as Social Security Number in a table with no more than one record per person) or it can be generated by the DBMS (such as a globally unique identifier, or GUID, in Microsoft SQL Server). Primary keys may consist of a single attribute/column or multiple attributes/columns in combination. No column that is part of the primary key can contain a null value.
- **Foreign Key**: A field or set of fields in one table that is indexed in another key; they refer to the primary key of another table and provide the building blocks of relating tables in a relational database. While primary keys must contain unique values, foreign keys may have duplicates.



Reporting Tables – Data Dictionaries

Table: PS_UHIR_STUDENTS

Record Name: **PS_UHIR_STUDENTS**

Record Description: This record contains **current information** for all students that have been previously enrolled in a class. Only one record exists per student, per institution. If a student has ever been enrolled at two institutions, two records will exist for that student.

Field Names	Field Descriptions
EMPLID (Key)	The unique PeopleSoft ID assigned to the student.
INSTITUTION (Key)	The specific institution that the student for which the student term activated.
UHIR_ENROLLED_CURR	Indicates if the student is currently enrolled (Y = Yes and N = No).
NAME	The full name of the student (Format Lastname, Firstname Middle.
LAST_NAME	The last name of the student.
FIRST_NAME	The first name of the student.
MIDDLE_NAME	The middle name of the student.
UHIR_USER_ID	The student's UH assigned user ID (i.e. HA).
FERPA	Indicates if the student wants personal data withheld from the public.
BIRTHDATE	The student's date of birth.
SEX	The student's gender.
ETHNIC_GRP_CD	The student's primary ethnic/race group for the USA regulatory region in PS prior to fall 2010.
UHCB_ETHNIC_ORIGIN	The students ethnic/race group as reported on the CB reports prior to fall 2010 (includes codes 1 thru 7)
UHIR_NEW_ETHNIC_CB	IR CB New Ethnicity starting Fall 2010 and after (includes codes 1 thru 9)
UHIR_NEW_EDESC_CB	IR CB New Ethnic Description starting Fall 2010 and after
EMAIL_ADDR	The UH email address for the student.
UHIR_OTHR_EMAIL	Other Email Address
UHIR_DEST_EMAIL	Destination Email
PHONE	The students home phone number.
UHIR_MAIL_ADDRESS1	The student's latest mailing address street 1 line.
UHIR_MAIL_ADDRESS2	The student's latest mailing address street 2 line.
UHIR_MAIL_ADDRESS3	The student's latest mailing address street 3 line.
UHIR_MAIL_ADDRESS4	The student's latest mailing address street 4 line.
UHIR_MAIL_CITY	The student's latest mailing city
UHIR_MAIL_STATE	The student's latest mailing state
UHIR_MAIL_POSTAL	The student's latest mailing zip code
UHIR_MAIL_COUNTRY	The student's latest mailing country
UHIR_MAIL_COUNTY	The student's latest mailing county
UHIR_PERM_ADDRESS1	The student's latest mailing address street 1 line.
UHIR_PERM_ADDRESS2	The student's latest permanent address street 2 line.
UHIR_PERM_ADDRESS3	The student's latest permanent address street 3 line.
UHIR_PERM_ADDRESS4	The student's latest permanent address street 4 line.
UHIR_PERM_CITY	The student's latest permanent city

UHIR_PERM_STATE	The student's latest permanent state
UHIR_PERM_POSTAL	The student's latest permanent zip code
UHIR_PERM_COUNTRY	The student's latest permanent country
UHIR_PERM_COUNTY	The student's latest permanent county
UHIR_HOME_ADDRESS1	The student's latest home address street 1 line.
UHIR_HOME_ADDRESS2	The student's latest home address street 2 line.
UHIR_HOME_ADDRESS3	The student's latest home address street 3 line.
UHIR_HOME_ADDRESS4	The student's latest home address street 4 line.
UHIR_HOME_CITY	The student's latest home city
UHIR_HOME_STATE	The student's latest home state
UHIR_HOME_POSTAL	The student's latest home zip code
UHIR_HOME_COUNTRY	The student's latest home country
UHIR_HOME_COUNTY	The student's latest home county
POS_SRVC_INDICATOR	Flag indicates if the student currently has at least one positive indicator
NEG_SRVC_INDICATOR	Flag indicates if the student currently has at least one negative indicator
UHIR_FIRSTTERMINST	First term enrolled in the Institution
UHIR_FTIC_STRM	First term a first-time in college student enrolled in the institution
UHIR_UGTRNSFR_STRM	First term an undergraduate transfer student enrolled in the institution
UHIR_GRAD_STRM	First term graduate student enrolled in the institution
UHIR_SP_STRM	First term special professional student enrolled in the institution
UHIR_PB_STRM	First term a student enrolled in the institution as a post-baccalaureate
UHIR_ETHNIC_WHITE	Student chose White as his/her race.
UHIR_ETHNIC_BLACK	Student chose African American or Black.
UHIR_ETHNIC_HISPA	Student indicated he/she was Hispanic
UHIR_ETHNIC_ASIAN	Student chose Asian American as his/her race
UHIR_ETHNIC_NSPEC	Student did not specify a race.
UHIR_ETHNIC_AMIND	Student chose Native American as his/her race.
UHIR_ETHNIC_PACIF	Student chose Hawaiian Pacific Islander as his/her race

Table: PS_UHIR_STDNT_STRM

Record Name:	**PS_UHIR_STDNT_STRM**
Record Description:	This record contains the student's semester specific information. Only students that enrolled for at least one class for a specific semester are included.

Field Names	Field Descriptions
EMPLID (Key)	The unique PeopleSoft ID assigned to the student.
INSTITUTION (Key)	The institution for which the student's transfer hours were applied.
ACAD_CAREER (Key)	The academic career for which the student is associated.
STRM (Key)	The University of Houston System semester term.
STDNT_CAR_NBR (Key)	The distinct PeopleSoft student career number assigned by PeopleSoft.
UHIR_ENROLL_CENSUS	Indicates if the student was enrolled in at least one class as of the official census date.
UHIR_ENROLL_CURR	Indicates if the student is currently enrolled. The student must be enrolled in at least one class. If the student drops his last class, the flag is set to 'N'. This flag will be set to 'N' if the semester has been completed.
UHIR_ENROLL_EOS	Indicates if the student was enrolled in at least one class on the last day of the semester. This flag will be set to 'N' if the semester has not been completed.
ACAD_LEVEL_BOT	The academic level code (i.e. 30, 40). These codes indicate if a student is a "Junior", "Senior"….
UHIR_ACAD_LEVEL	The academic level description (i.e. Junior, Senior) corresponding to the academic level codes.
~~UHIR_COLLEGE~~	not used
UHIR_COLLEGE_DESCR	not used
UHIR_PRIMARY_ORG	not used
PRIMARY_ORG_DESCR	not used
UHIR_ACAD_STANDING	Current Term Academic Standing- indicates standing for the STRM
UHIR_ACAD_STNDDESC	Current Acad Standing Desc - indicates the standing description for the STRM
UHIR_PRE_ACAD_STD	Prior semester academic standing - Indicates standing for the last enrolled STRM
UHIR_P_ACAD_STDESC	Prior Term Acad Stdg Descr - Indicates standing description for the last enrolled STRM
UHIR_P_ACDSTD_TERM	Prior Acad Standing Term - Indicates the STRM for the last enrolled standing
~~ACAD_STANDING~~	No longer Available
~~ACAD_PROG_PRIMARY~~	not used
EXP_GRAD_TERM	Expected Graduation Term
~~UHIR_PRIMARY_PLAN~~	not used
UHIR_PLAN_DESC	not used
UHIR_SUB_PLAN1	not used
UHIR_SUB_PLAN1DESC	not used
UHIR_SUB_PLAN2	not used
UHIR_SUB_PLAN2DESC	not used
UHIR_SUB_PLAN3	not used
UHIR_SUB_PLAN3DESC	not used
DECLARE_DT	Declare Date
REQ_TERM	Requirement Term
DEGREE	Degree
CIP_CODE	CIP Code
~~UHIR_PLAN_TYPEDESC~~	not used
UHIR_MAJOR2	not used
UHIR_MINOR1	not used
UHIR_PLANDESC_MIN1	not used
CIP_CODE_MIN1	not used
~~UHIR_MINOR2~~	not used

UHIR_FIRST_TERM	The student's first semester in the specific institution and academic career.
UHIR_LAST_TERM	The student's last semester in the specific institution and academic career.
ADMIT_TERM	The term that student was admitted into a specific institution and acad career. This date can change after the student is originally admitted for various reasons.
CUM_GPA	The student's cumulative GPA for all semesters in the specific institution and career.
CUR_GPA	The student's GPA for one semester. This table is keyed on Institution, Term and Career, so the value in this field is the GPA for the one semester for the one career in the one institution.
GRADE_POINTS	The number of grade points earned.
TOT_CUMULATIVE	Total Cumulative Units
TOT_GRADE_POINTS	Total Grade Points
TOT_INPROG_GPA	The student's GPA for courses that are in progress.
TOT_INPROG_NOGPA	Total In Progress -Not for GPA
TOT_PASSD_FA	Total Fin Aid Units Passed
TOT_PASSD_GPA	The student's GPA for courses passed.
TOT_PASSD_NOGPA	Total Passed Not Toward GPA
TOT_PASSD_PRGRSS	Total Passed for Progress
TOT_TAKEN_FA	Total Fin Aid Units Taken
TOT_TAKEN_FA_GPA	Total Fin Aid Taken Toward GPA
TOT_TAKEN_GPA	The student's GPA for courses taken.
TOT_TAKEN_NOGPA	Total Taken Not Toward GPA
TOT_TAKEN_PRGRSS	Total Taken for Progress
TOT_TEST_CREDIT	Total From Test Credit
TOT_TRNSFR	Total Transferred
TRF_GRADE_POINTS	Transfer Grade Points
TRF_PASSED_GPA	Transfer Passed for GPA
TRF_PASSED_NOGPA	Transfer Passed Not for GPA
TRF_TAKEN_GPA	Transfer Taken for GPA
TRF_TAKEN_NOGPA	Transfer Taken Not for GPA
UNT_INPROG_GPA	Units In Progress - GPA
UNT_INPROG_NOGPA	Unit In Progress - Not for GPA
UNT_PASSD_FA	Fin Aid Progress Units Passed
UNT_PASSD_GPA	Units Passed Toward GPA
UNT_PASSD_NOGPA	Units Passed Not Toward GPA
UNT_PASSD_PRGRSS	Units Passed for Progress
UNT_TAKEN_FA	Fin Aid Progress Units Taken
UNT_TAKEN_FA_GPA	FA Units Taken Toward GPA
UNT_TAKEN_GPA	Units Taken Toward GPA
UNT_TAKEN_NOGPA	Units Taken Not Toward GPA
UNT_TAKEN_PRGRSS	Units Taken for Progress
UNT_TERM_TOT	Total Term Units
UNT_TEST_CREDIT	Units from Test Credit
UNT_TRNSFR	Units Transferred
UHIR_DEANS_LIST	Flag (Y or N) indicates if student made the Dean's List during the specific semester.
RESIDENCY	Residency
TUITION_RES	Tuition Residency
TUITION_EXCPT	Tuition Residency Exception
UHIR_US_CITSTATUS	US Citizenship Status
UHIR_CITIZ_COUNTRY	Citizen Country
UHIR_NONUS_CITSTAT	Non USA Citizen
VISA_PERMIT_TYPE	Visa/Permit Type

UHIR_VISA_EFFDT	Visa Effective Date
EXPIRATN_DT	Visa Expiration Date
BIRTHCOUNTRY	Birth Country
ELIG_TO_ENROLL	Eligible to Enroll
ACADEMIC_LOAD	Academic Load
WITHDRAW_DATE	Withdrawal \ Cancel Date
WITHDRAW_CODE	Withdrawal \ Cancel
WITHDRAW_REASON	Withdrawal \ Cancel Reason
UHIR_ACAD_GRP_1	Academic Group 1
UHIR_GRP_DESCR_1	Academic Group 1 Description
UHIR_ACAD_ORG_1	Academic Org 1
UHIR_ORG_DESCR_1	Academic Org 1 Description
UHS_PRIM_PROG_1	Academic Program 1
UHS_PROG_DESCR_1	Academic Program 1 Description
UHS_PRIM_PLAN_1A	Academic Plan 1A
UHS_PLAN_DESCR_1A	Academic Plan 1A Description
UHS_PRIM_PLAN_1B	Academic Plan 1B
UHS_PLAN_DESCR_1B	Academic Plan 1B Description
UHS_PRIM_PLAN_1C	Academic Plan 1C
UHS_PLAN_DESCR_1C	Academic Plan 1C Description
UHS_PRIM_MINOR_1A	Minor Plan 1A
UHS_MINOR_DESCR_1A	Minor Plan 1A Description
UHS_PRIM_MINOR_1B	Minor Plan 1B
UHS_MINOR_DESCR_1B	Minor Plan 1B Description
UHS_PRIM_MINOR_1C	Minor Plan 1C
UHS_MINOR_DESCR_1C	Minor Plan 1C Description
UHIR_ACAD_GRP_2	Academic Group 2
UHIR_GRP_DESCR_2	Academic Group 2 Description
UHIR_ACAD_ORG_2	Academic Org 2
UHIR_ORG_DESCR_2	Academic Org 2 Description
UHS_PROG_2	Academic Program 2
UHS_PROG_DESCR_2	Academic Program 2 Description
UHS_PLAN_2A	Academic Plan 2A
UHS_PLAN_DESCR_2A	Academic Plan 2A Description
UHS_PLAN_2B	Academic Plan 2B
UHS_PLAN_DESCR_2B	Academic Plan 2B Description
UHS_PLAN_2C	Academic Plan 2C
UHS_PLAN_DESCR_2C	Academic Plan 2C Description
UHS_MINOR_2A	Minor Plan 2A
UHS_MINOR_DESCR_2A	Minor Plan 2A Description
UHS_MINOR_2B	Minor Plan 2B
UHS_MINOR_DESCR_2B	Minor Plan 2B Description
UHS_MINOR_2C	Minor Plan 2C
UHS_MINOR_DESCR_2C	Minor Plan 2C Description
UHIR_ACAD_GRP_3	Academic Group 3
UHIR_GRP_DESCR_3	Academic Group 3 Description
UHIR_ACAD_ORG_3	Academic Org 3
UHIR_ORG_DESCR_3	Academic Org 3 Description
UHS_PROG_3	Academic Program 3
UHS_PROG_DESCR_3	Academic Program 3 Description
UHS_PLAN_3A	Academic Plan 3A

UHS_PLAN_DESCR_3A	Academic Plan 3A Description
UHS_PLAN_3B	Academic Plan 3B
UHS_PLAN_DESCR_3B	Academic Plan 3B Description
UHS_PLAN_3C	Academic Plan 3C
UHS_PLAN_DESCR_3C	Academic Plan 3C Description
UHS_MINOR_3A	Minor Plan 3A
UHS_MINOR_DESCR_3A	Minor Plan 3A Description
UHS_MINOR_3B	Minor Plan 3B
UHS_MINOR_DESCR_3B	Minor Plan 3B Description
UHS_MINOR_3C	Minor Plan 3C
UHS_MINOR_DESCR_3C	Minor Plan 3C Description
UHS_SUBPLAN_PLAN_1	Subplan Plan 1
UHS_SUBPLAN_1	Subplan 1
UHS_SUBPLAN_DESCR1	Subplan 1 Description
UHS_SUBPLAN_PLAN_2	Subplan Plan 2
UHS_SUBPLAN_2	Subplan 2
UHS_SUBPLAN_DESCR2	Subplan 2 Description
UHS_SUBPLAN_PLAN_3	Subplan Plan 3
UHS_SUBPLAN_3	Subplan 3
UHS_SUBPLAN_DESCR3	Subplan 3 Description
UHS_ADTNL_PROGRAM	Indicates whether the student has additional programs not included in the table.
UHIR_ACTIVE_PROG	Indicates whether the plan information is based on an active plan or inactive plan.
UHIR_MILLIT_STATUS	Millitary Status-Veteran, Dependent, Active Duty
UHIR_VA_BEN_ELIGBL	VA Benefits Eligible
UHIR_HZLWD_BENRECV	Recevied Hazlewwod Benefit

Table: PS_UHIR_STDNT_ENRL

Record Name: **PS_UHIR_STDNT_ENRL**

Record Description: This record contains the student's class enrollment data by semester

Field Names	Field Descriptions
EMPLID (Key)	The unique PeopleSoft ID assigned to the student.
INSTITUTION (Key)	The institution for which the student's enrollment applies.
ACAD_CAREER (Key)	The academic career for which the student's enrollment applies.
STRM (Key)	The semester of the enrollment.
CLASS_NBR (Key)	The class number.
UHIR_COURSE_DESCR	The description of the course.
CRSE_CAREER	The course career.
CRSE_ID	Course ID
CRSE_OFFER_NBR	Course Offering Nbr
SSR_COMPONENT	Course Component - Type of Course (i.e. Lecture, Lab….)
SESSION_CODE	The session code.
SESSN_ENRL_CNTL	Enrollment Control Session (i.e. Regular Academic Session or 6 Week - First).
STDNT_ENRL_STATUS	Enrollment Status (i.e. Enrolled or Dropped). The Enrollment Status stay 'E' after census.
ENRL_STATUS_REASON	Reason for Enrollment Status (i.e. WDRW - Withdrew).
ENRL_ACTION_LAST	Last Enrollment Action (i.e. S = Swap Courses).
ENRL_ACTN_RSN_LAST	Reason for Last Enrollment Action.
ENRL_ACTN_PRC_LAST	Last Enrollment Action Process.
STATUS_DT	Status Date
ENRL_ADD_DT	Date student enrolled in class
ENRL_DROP_DT	Date student dropped class.
UNT_TAKEN	Units taken.
UNT_PRGRSS	Units taken for academic progress.
CRSE_COUNT	Course credit count.
CRSE_GRADE_OFF	Official grade.
CRSE_GRADE_INPUT	The grade that was input for the course.
GRADE_DT	The effective date of the grade.
STDNT_POSITIN	Student Position
EARN_CREDIT	Earn Credit
UNT_EARNED	Units Earned
INCLUDE_IN_GPA	Include class grade in GPA
UNITS_ATTEMPTED	Units Attempted (i.e. Yes, No or In Progress)
GRADE_POINTS	Grade Points
GRD_PTS_PER_UNIT	Grade Points per unit.
INSTRUCTOR_ID	Instructor ID.
INSTR_NAME	Instructor
INSTRUCTION_MODE	Method of Instruction (i.e. Face to Face, Internet, Video Tape).
GRADE_CATEGORY	Grade Category.
ENRL_REQ_SOURCE	Source of enrollment request (i.e. Self-Service Enrollment, Mass Enrollment….)
LAST_UPD_ENREQ_SRC	Source for last time enrollment was updated (i.e. Grade Posting, Self-Service Enrollment…)
ACAD_PROG	Academic program of the course.
SUBJECT	Course subject.
CATALOG_NBR	Catalog number.
CLASS_SECTION	Class section.
ENRL_CAP	Enrollment capacity in the class.
ENRL_TOT	Total enrolled in class.
START_DT	First day of class.
END_DT	Last day of class.
CAMPUS	Campus for class.
LOCATION	Location of class.
UHIR_HONORS_FLG	Honors Flag

Table: PS_UHIR_STDNT_DEGR

Record Name: **PS_UHIR_STDNT_DEGR**

Record Description: This record contains student's degree information (if applicable).

Field Names	Field Descriptions
EMPLID (Key)	The unique PeopleSoft ID assigned to the student.
STDNT_DEGR (Key)	Distinct degree number assigned by PeopleSoft each time a student obtains a degree. If a student has three degrees, each degree will have a distinct student degree number (i.e. 01, 02, 03).
EXT_ORG_ID (Key)	The distinct external institution that the student earned a degree. This field is blank when the student earned the degree at a University of Houston System institution.
EXT_DEGREE_NBR (Key)	Distinct degree number assigned by PeopleSoft each time a students non-UH (external) degree information is loaded. If a student has three external degrees, each degree will have distinct external degree number (i.e. 1, 2, 3).
UHIR_UH_DEGREE_FLG	This flag indicates if the degree was obtained at U of H. If so, the flag is set to 'Y'.
UHIR_DEGR_CAREER	The career for which a degree is associated (i.e. Law, Pharmacy, Graduate).
UHS_DEGREE_INST	Degree UH Institution
UHIR_DEGR_INST	The description of the institution for which the degree was earned.
UHIR_COLLEGE	College
UHIR_PRIMARY_ORG	Primary Plan Org Owner
DEGREE	The degree code.
UHIR_DEGR_DESCR	The description for the type of degree earned (i.e. High School Diploma, Bachelor of Arts).
UHIR_DEGR_DT	The date the degree was earned.
ACAD_DEGR_STATUS	The academic degree status.
GPA_DEGREE	The GPA for obtained seeking the degree.
COMPLETION_TERM	The semester the degree requirements were completed.
UHIR_DEGR_HONORS	The honors associated to the degree (i.e. Cum Laude, Magna Cum Laude).
UHIR_DEGR_MAJOR1	The primary major associate to the degree earned (i.e. Accounting, BBA, Psychology, BS).
UHIR_DEGR_MAJ1SUB1	The first subplan associated with the Major1 degree.
UHIR_DEGR_MAJ1SUB2	The second subplan associated with the Major1 degree.
UHIR_DEGR_MAJOR2	The secondary major associate to the degree earned (i.e. Finance, BBA, Marketing, BBA).
UHIR_DEGR_MAJ2SUB1	The first subplan associated with the Major2 degree.
UHIR_DEGR_MAJ2SUB2	The second subplan associated with the Major2 degree.
UHIR_DEGR_MINOR1	The primary minor associate to the degree earned (i.e. Psychology, Chemistry).
UHIR_DEGR_MIN1SUB1	The first subplan associated with the Minor1 degree.
UHIR_DEGR_MIN1SUB2	The second subplan associated with the Minor 1 degree.
UHIR_DEGR_MINOR2	The secondary minor associate to the degree earned (i.e. Spanish, Mathematics).
UHIR_DEGR_MIN2SUB1	The first subplan associated with the Minor 2 degree.
UHIR_DEGR_MIN2SUB2	The second subplan associated with the Minor 2 degree.

Table: PS_UHIR_STDNT_TRAN

Record Name: **PS_UHIR_STDNT_TRAN**

Record Description: This record contains the student's transfer information (if applicable).

Field Names	Field Descriptions
EMPLID (Key)	The unique PeopleSoft ID assigned to the student.
INSTITUTION (Key)	The institution for which the student's transfer hours were applied.
ACAD_CAREER (Key)	The academic career for which the student's transfer hours were applied.
EXT_GPA	External GPA (equal to institution 1 GPA, if not equal to zero, otherwise equal to institution 2 GPA if not equal to zero, otherwise equal to institution 3 GPA if not equal to zero.
UH_ESEM_LAST1	The last semester the student attended the most recent transfer institution.
UH_ESEM_LAST2	The last semester the student attended the second most recent transfer institution.
UH_ESEM_LAST3	The last semester the student attended the third most recent transfer institution.
UH_ESEM_LAST4	The last semester the student attended the fourth most recent transfer institution.
UH_ESEM_LAST5	The last semester the student attended the fifth most recent transfer Institution.
UHIR_BSEM_LAST1	The first semester the student attended the most recent transfer institution.
UHIR_BSEM_LAST2	The first semester the student attended the second most recent transfer institution.
UHIR_BSEM_LAST3	The first semester the student attended the third most recent transfer institution.
UHIR_BSEM_LAST4	The first semester the student attended the fourth most recent transfer institution.
UHIR_BSEM_LAST5	The first semester the student attended the fifth most recent transfer institution.
UHIR_INST1_GPA	GPA at the student's most recent transfer institution.
UHIR_INST2_GPA	GPA at the student's second most recent transfer institution.
UHIR_INST3_GPA	GPA at the student's third most recent transfer institution.
UHIR_INST4_GPA	GPA at the student’s fourth most recent transfer institution.
UHIR_INST5_GPA	GPA at the student’s fifth most recent transfer institution.
UHIR_INST1_NAME	The name of the student's most recent transfer institution.
UHIR_INST2_NAME	The name of the student's second most recent transfer institution.
UHIR_INST3_NAME	The name of the student's third most recent transfer institution.
UHIR_INST4_NAME	The name of the student’s fourth most recent transfer institution.
UHIR_INST5_NAME	The name of the student’s fifth most recent transfer institution.
UHIR_INST1_FICE	The FICE Code for the student's most recent transfer institution.
UHIR_INST2_FICE	The FICE Code for the student's second most recent transfer institution.
UHIR_INST3_FICE	The FICE Code for the student's third most recent transfer institution.
UHIR_INST4_FICE	The FICE Code for the student’s fourth most recent transfer institution.
UHIR_INST5_FICE	The FICE Code for the student’s fifth most recent transfer institution.
UHIR_XFR_HRS_TKN1	Transfer hours taken at student's most recent institution.
UHIR_XFR_HRS_TKN2	Transfer hours taken at student's second most recent institution.
UHIR_XFR_HRS_TKN3	Transfer hours taken at student's third most recent institution.
UHIR_XFR_HRS_TKN4	Transfer hours taken at student’s fourth most recent institution.

UHIR_XFR_HRS_TKN5	Transfer hours taken at student’s fifth most recent institution.
UHIR_XFR_HRS_ERN1	Hours transferred from the student's most recent institution.
UHIR_XFR_HRS_ERN2	Hours transferred from the student's second most recent institution.
UHIR_XFR_HRS_ERN3	Hours transferred from the student's third most recent institution.
UHIR_XFR_HRS_ERN4	Hours transferred from the student’s fourth most recent institution.
UHIR_XFR_HRS_ERN5	Hours transferred from the student’s fifth most recent institution.
UHIR_XFR_HRS_TKN	Total hours taken at all transfer institutions.
UHIR_XFR_HRS_ERN	Total hours transferred from all transfer institutions.

Table: PS_UHIR_STDNT_ACTI

Record Name: **PS_UHIR_STDNT_ACTI**

Record Description: This record contains student's activity information (if applicable). This record combines student attributes, extra curricular, student activity, and student group data.

Field Names	Field Descriptions
EMPLID (Key)	PeopleSoft ID
INSTITUTION (Key)	Academic Institution
ACAD_CAREER (Key)	Academic Career
STRM (Key)	Term
STDNT_CAR_NBR (Key)	Student Career Nbr
UHIR_ACTIVITY_TYPE (Key)	Student Activity Type
UHIR_ACTIVITY (Key)	Student Activity
UHIR_ACT_TYPEDESCR	Activity Type Description
UHIR_ACTIVITYDESCR	Student Activity Description
START_DT	Start Date
END_DT	End Date
EXTRA_ACTVTY_PRIM	Extra Activity Primacy
STDNT_ATTR_VAL_PRM	Attribute Value Primacy
NCAA_ELIGIBLE	NCAA Eligible
CUR_PARTICIPANT	Current Participant
UHIR_ACTIVITY_SRC	Activity Source Data

Table: PS_UHIR_CLASS_TBL

Record Name: **PS_UHIR_CLASS_TBL**

Record Description: This reporting table record contains data related to the classes offered at one of the U of H institutions.

Field Names	Field Descriptions
CRSE_ID (Key)	Course ID
CRSE_OFFER_NBR (Key)	Course Offering Nbr
STRM (Key)	The semester (term) for which the class data is associated.
SESSION_CODE (Key)	The session code (i.e. Regular or 8 weeks first)
CLASS_SECTION (Key)	The class section
INSTITUTION	The institution that offers the course
CLASS_NBR	The unique class number.
UHIR_COURSE_DESCR	The description of the course
UHCB_004_FLG	Is the class found in the CB004 report ("Y" = Yes and "N" = No)
UHIR_CRSE_1ST_TERM	The first semester (term) the course was offered
UHIR_CRSE_LASTTERM	The last semester (term) the course was offered
UHIR_FIRST_YEAR	The first year the course was offered
UHIR_LAST_YEAR	The last year the course was offered
UHIR_LAST_ENRLTERM	The last semester (term) that a student enrolled in the class.
UHIR_LAST_ENRLYEAR	The last year that a student enrolled in the class.
ACAD_GROUP	The academic group (college) that offers the course
SUBJECT	The course subject
CATALOG_NBR	The unique catalog number for the course
SSR_COMPONENT	The component (i.e. Lecture, Practicum)
ASSOCIATED_CLASS	Associated class
CLASS_TYPE	The type of class
CLASS_STAT	Class Status
CRS_TOPIC_ID	Course Topic ID
CRSE_TOPIC_DESCR	Course Topic Description
INSTRUCTION_MODE	The mode used for instructing the class.
UH_INSTRMODE_DESCR	Instruct Mode Description
START_DT	The first day of class.
END_DT	The last day of class.
CAMPUS	The campus that course is offered
LOCATION	The location of the class.
FACILITY_ID	Facility ID
BLDG_CD	Building
UHIR_BLDG_DESCR	Building Description
ROOM	Room
UHIR_ROOM_DESCR	UHIR_ROOM_DESCR
ROOM_CAPACITY	Room Capacity
ACAD_ORG	The department that offers the course
ACAD_CAREER	The career for the course
CIP_CODE	The CIP Code for the course.
UNITS_MAXIMUM	Maximum units earned by taking the class.
UNITS_MINIMUM	Minimum Units
GRADING_BASIS	The grading basis used for the class.
ROOM_CAP_REQUEST	The capacity of the room in which the course is taught. How many student could sit in the classroom
ENRL_CAP	The quota or total number students allowed to enrolled.
ENRL_TOT	The total number of students that enrolled in the class.
ENRL_STAT	The enrollment status (i.e. Open or Closed)
UHIR_OFF_CAMP_CITY	The city that the class is offered if not at UH.
UHIR_OFF_CAMP_ZIP	The zipcode that the class is offered if not at UH.
UHIR_OFF_CAMP_CNTR	The country that the class is offered if not at UH.

UHCB_LAB_HRS	The number of lab hours for the class.
UHCB_LEC_HRS	The number of lecture hours for the class.
INSTRUCTOR_ID	The primary instructor ID for the class.
INSTR_NAME	The name of the primary instructor for the class.
UHCB_RANK	The instructor rank based on the CB report.
UHIR_SCHED_DAYTIME	Scheduled Time and Day

Table: PS_UHIR_ADMISSIONS

Record Name: **PS_UHIR_ADMISSIONS**

Record Description: This record contains student's application information.

Field Names	Field Descriptions
INSTITUTION (Key)	Institution
ADMIT_TERM (Key)	Admission Term
EMPLID (Key)	Student's PS ID
ACAD_CAREER (Key)	Academic Career
STDNT_CAR_NBR (Key)	Student's Career Number
ADM_APPL_NBR (Key)	Application Number
APPL_PROG_NBR (Key)	Application Program Number
UHIR_CAREER_GRP	Career Group ('UG' – undergraduate, 'GR' – graduate, 'SP' – pharmacy, optometry, law)
UHIR_ADM_TERM_DESC	Admit Term Description
NATIONAL_ID	SSN (not populated)
NAME	Student's Name (Last, First Middle)
FIRST_NAME	Student's First Name
MIDDLE_NAME	Student's Middle Name
LAST_NAME	Student's Last Name
BIRTHDATE	Birth Date
SEX	Gender (M / F)
ETHNIC_GRP_CODE	Ethnic Group Code - Prior to Fall 2010
UHIR_NEW_ETHNIC	IR New Ethnicity - Fall 2010 and after
UHIR_NEW_EDESC	IR New Ethnic Description - Fall 2010 and after
UHIR_ETHNIC_WHITE	UHIR Ethnicity White PS - Fall 2010 and after
UHIR_ETHNIC_BLACK	UHIR Ethinicity Black PS - Fall 2010 and after
UHIR_ETHNIC_HISPA	UHIR Ethnicity Hispa PS - Fall 2010 and after
UHIR_ETHNIC_ASIAN	UHIR Ethnicity Asian PS - Fall 2010 and after
UHIR_ETHNIC_NSPEC	UHIR Ethnicity Nspec PS - Fall 2010 and after
UHIR_ETHNIC_AMIND	UHIR Ethnicity AmeInd PS - Fall 2010 and after
UHIR_ETHNIC_PACIF	UHIR Ethnicity Pacif Island PS - Fall 2010 and after
VISA_PERMIT_TYPE	Visa Type
RESIDENCY	Residency
RESIDENCY_DT	Residency Date
EFFECTIVE_TERM	Residency Effective Term
TUITION_RES	Tuition Residency
TUITION_EXCPT	Tuition Residency Exception
ADMIT_TYPE	Admissions Type
UHIR_ADM_TYPE_DESC	Admissions Type Description
PROG_ACTION	Program Action Code
ACTION_DT	Action Date
PROG_REASON	Action Reason
PROG_STATUS	Academic Program Status
ACTION_DESCRSHORT	Program Action Description
UHIR_ADM_ACCEPTED	Student Accepted for Admission (Y / N)
UHIR_ADM_DENIED	Admission Denied from Admission (Y / N)
ENROLLED	Student Enrolled (Y / N)
UHIR_PERM_ADDRESS1	Address Line 1 (Primary Address)
UHIR_PERM_ADDRESS2	Address Line 2 (Primary Address)
UHIR_PERM_ADDRESS3	Address Line 3 (Primary Address)
UHIR_PERM_ADDRESS4	Address Line 4 (Primary Address)
UHIR_PERM_CITY	City (Primary Address)

UHIR_PERM_POSTAL	Zip Code (Primary Address)
UHIR_PERM_STATE	State (Primary Address)
UHIR_PERM_COUNTRY	Country (Primary Address)
UHIR_PERM_COUNTY	County (Primary Address)
UHIR_MAIL_ADDRESS1	Mail Address 1
UHIR_MAIL_ADDRESS2	Mail Address 2
UHIR_MAIL_ADDRESS3	Mail Address 3
UHIR_MAIL_ADDRESS4	Mail Address 4
UHIR_MAIL_CITY	Mail City
UHIR_MAIL_POSTAL	Mail Postal-Zipcode
UHIR_MAIL_STATE	Mail State
UHIR_MAIL_COUNTRY	Mailing Address Country
UHIR_MAIL_COUNTY	Mailing Address County
EMAIL_ADDR	Student's Email Address (Preferred)
UHIR_UH_EMAIL	UH Email Address
UHIR_OTHR_EMAIL	Other Email Address
PHONE	Phone Number (Preferred)
MAR_STATUS	Marital Status
FERPA	Privacy Code
UHIR_HS_ORG_ID	High School's External Organization ID
UHIR_HS_ATP_CD	High School's ATP Code
UHIR_HS_FICE_CD	High School's FICE Code
UHIR_HS_NAME	High School's Name
UHIR_HS_STATE	High School's State
UHIR_HS_GRAD_DT	High School Graduation Date
UHIR_HS_GRAD_YR	High School Graduation Year
UHIR_HS_DEGREE	High School Degree Code
UHIR_HS_DEGR_DESCR	High School Degree Description
UHIR_HS_GPA	High School GPA
UHIR_HS_CLASS_SIZE	High School Class Size
UHIR_HS_CLASS_RANK	High School Class Rank
UHIR_HS_PERCENTILE	High School Class Percentile
UHIR_HS_QUARTILE	High School Class Quartile
UHIR_HS_SUMM_TYPE	High School Sum Type
UHIR_HS_TERM_YEAR	High School Term Year
ACAD_GROUP	Academic Group
UHIR_ACAD_GRP_DESC	Academic Group Description
ACAD_ORG	Academic Organization
UHIR_ACAD_ORG_DESC	Academic Organization Description
ACAD_PROG	Academic Program
ACAD_PLAN	Academic Plan (Major or Highest Plan Sequence Number)
UHIR_PLAN_DESC	Academic Plan Description
UHIR_PLAN_DEGREE	Academic Plan Degree
UHIR_PLAN_DEG_DESC	Academic Plan Degree Description
UHIR_ACAD_PLAN2	Second Academic Plan (Highest Plan Sequence Number, if Major not highest, or Second Highest Sequence Number)
UHIR_PLAN2_DESC	Second Academic Plan Description
UHIR_ACAD_PLAN3	Third Academic Plan (Next Highest Plan Sequence Number, after Second Academic Plan)
UHIR_PLAN3_DESC	Third Academic Plan Description
UHIR_SUBPLAN1	Academic Sub-Plan 1
UHIR_SUBPLAN_D1	Sub-Plan 1 Description
UHIR_SUBPLAN2	Academic Sub-Plan 2

UHIR_SUBPLAN_D2	Sub-Plan 2 Description
UHIR_PLAN2_SPLAN1	Academic Sub-Plan 1
UHIR_PLAN2_SPLAND1	Sub-Plan 1 Description
UHIR_PLAN2_SPLAN2	Academic Sub-Plan 2
UHIR_PLAN2_SPLAND2	Sub-Plan 2 Description
UHIR_PLAN3_SPLAN1	Academic Sub-Plan 1
UHIR_PLAN3_SPLAND1	Sub-Plan 1 Description
UHIR_PLAN3_SPLAN2	Academic Sub-Plan 2
UHIR_PLAN3_SPLAND2	Sub-Plan 2 Description
UHIR_ADMDES_DT	Admission Decision Date
ADM_APPL_DT	Application Date
AGE	Student's Age
UHIR_ATHLETIC_REC	Athletic Recruit (Y / N)
UHIR_HONORS_REC	Honors Recruit (Y / N)
ACAD_LOAD_APPR	Approved Academic Load
EXT_GPA	External GPA (equal to institution 1 GPA, if not equal to zero, otherwise equal to institution 2 GPA if not equal to zero, otherwise equal to institution 3 GPA if not equal to zero.
UHS_TOTAL_GPA	Total GPA
UHIR_XFR_HRS_TKN	Total Hours to be Transferred
UHIR_XFR_HRS_ERN	Total Hours Allowed for Transfer
UHIR_BASIS_ADMIT	Basis of Admission Description
UHIR_PROG_RSN_DESC	Program Reason Code Description
UHIR_RECRUIT_CAT1	Recruitment Category Description 1
UHIR_RECRUIT_CAT2	Recruitment Category Description 2
UHIR_RECRUIT_CAT3	Recruitment Category Description 3
UHIR_RECRUIT_CAT4	Recruiting Category 4
UHIR_NON_UH_INST	Last Non-UH Institution Attended
UHIR_STDNT_GRP1	Student Group Code Description 1
UHIR_STDNT_GRP2	Student Group Code Description 2
UHIR_STDNT_GRP3	Student Group Code Description 3
UHIR_ASSOC_DEG	Associate Degree Description
UHIR_ASSOC_REC	Associate Degree Received (Y / N)
UHIR_DEGREE_COUNT	Number of Degrees
ACADEMIC_LEVEL	Academic Level
UHIR_ACAD_LEVEL	Academic Level Description
BIRTHCOUNTRY	Birth Country
UHIR_CITZN_COUNTRY	Citizenship Country
UHIR_ESEM_LAST1	Last Semester at Previous Institution
UHIR_BSEM_LAST1	First Semester at Previous Institution
UHIR_INST1_GPA	GPA at Previous Institution
UHIR_INST1_NAME	Previous Institution Name
UHIR_INST1_FICE	Previous Institution FICE Code
UHIR_INST1_SUM_TYP	Last Inst 1 Sum Type
UHIR_INST1_TERM_YR	Last Inst 1 Term Year
UHIR_INST1_SCH_TYP	Last Inst 1 School Type
UHIR_XFR_HRS_TKN1	Transfer Hours Taken from the Previous Institution
UHIR_XFR_HRS_ERN1	Transfer Hours Allowed from the Previous Institution
UHIR_REC_DEG1	Degree Received from the Previous Institution
UHIR_DEG1_STRM	Term Degree Received from the Previous Institution
UHIR_ESEM_LAST2	Last Semester at Second Previous Institution
UHIR_BSEM_LAST2	First Semester at Second Previous Institution
UHIR_INST2_GPA	GPA at Second Previous Institution

UHIR_INST2_NAME	Second Previous Institution Name
UHIR_INST2_FICE	Second Previous Institution FICE Code
UHIR_INST2_SUM_TYP	Last Inst 2 Sum Type
UHIR_INST2_TERM_YR	Last Inst 2 Term Year
UHIR_INST2_SCH_TYP	Last Inst 2 School Type
UHIR_XFR_HRS_TKN2	Transfer Hours Taken from the Second Previous Institution
UHIR_XFR_HRS_ERN2	Transfer Hours Allowed from the Second Previous Institution
UHIR_REC_DEG2	Degree Received from the Second Previous Institution
UHIR_DEG2_STRM	Term Degree Received from the Second Previous Institution
UHIR_ESEM_LAST3	Last Semester at Third Previous Institution
UHIR_BSEM_LAST3	First Semester at Third Previous Institution
UHIR_INST3_GPA	GPA at Third Previous Institution
UHIR_INST3_NAME	Third Previous Institution Name
UHIR_INST3_FICE	Third Previous Institution FICE Code
UHIR_INST3_SUM_TYP	Last Inst 3 Sum Type
UHIR_INST3_TERM_YR	Last Inst 3 Term Year
UHIR_INST3_SCH_TYP	Last Inst 3 School Type
UHIR_XFR_HRS_TKN3	Transfer Hours Taken from the Third Previous Institution
UHIR_XFR_HRS_ERN3	Transfer Hours Allowed from the Third Previous Institution
UHIR_REC_DEG3	Degree Received from the Third Previous Institution
UHIR_DEG3_STRM	Term Degree Received from the Third Previous Institution
UHCB_FAT_LGL_EDUC	Father / Guardian Education Level
UHCB_MOT_LGL_EDUC	Mother / Guardian Education Level
UHCB_FAM_GRS_INV	Family Gross income
UHCB_NO_PPL_HOME	Number of People in Household
UHCB_ADDL_LNG_SPK1	Additional Language Spoken
UHCB_YEARS_SPOKEN1	Years Additional Language Spoken
UHCB_ADDL_LNG_SPK2	Second Additional Language Spoken
UHCB_YEARS_SPOKEN2	Years Second Additional Language Spoken
UHCB_FAM_OBLIG	Family Obligation
UHCB_FAM_OBLREASON	Family Obligation Reason
UHIR_HOUSE_INTD	Interested in Housing (Y / N)
FIN_AID_INTEREST	Interested in Financial Aid (Y / N)
UHIR_LST_CHKLST_CH	Date of Last Checklist Change
ADM_APPL_METHOD	Method of Application
STRM	Term (same as ADMIT_TERM)
ADM_APPL_METHOD	Application Method
UHS_TSI_STATUS	TSI Component Status
UHS_TSI_MATH_STAT	TSI Math Status
UHS_TSI_READ_STAT	TSI Read Status
UHS_TSI_WRITE_STAT	TSI Write Status
EXT_ADM_APPL_NBR	External Application Nbr
UHIR_PROG_EFFDT	Program Effective Date
ADM_CREATION_DT	Admission Data Creation Date
ALL_GROUPS	Flag Used for Selecting All Groups (Y)
ALL_CAREERS	Flag Used for Selecting All Careers (Y)
ALL_PLANS_CD	Flag Used for Selecting All Plans (Y)
ALL_EXT_ORG	Flag Used for Selecting All Organizations (Y)

Table: PS_UHIR_ADM_SCORES

Record Name: **PS_UHIR_ADM_SCORES**

Record Description: This record contains student's application test scores information.

Field Names	Field Descriptions
INSTITUTION (Key)	Institution
ADMIT_TERM (Key)	Admit Term
EMPLID (Key)	Student's PS ID
ACAD_CAREER (Key)	Academic Career
STDNT_CAR_NBR (Key)	Student's Career Number
ADM_APPL_NBR (Key)	Application Number
APPL_PROG_NBR (Key)	Application Program Nbr
UHIR_SAT_COMP	SAT Composite Score
UHIR_SAT_ENG	SAT Verbal Score (from Test with Highest Composite Score)
UHIR_SAT_MATH	SAT Math Score (from Test with Highest Composite Score)
UHIR_SAT_WRITE	SAT Writing Score (from Test with Highest Composite Score)
UHIR_SAT_CRIT	SAT Critical Reading Score (from Test with Highest Composite Score)
UHIR_SAT_TEST_DT	SAT Test Date
UHIR_SAT_ENG_MAX	Maximum SAT Verbal Score (from any Test)
UHIR_SAT_MATH_MAX	Maximum SAT Math Score (from any Test)
UHIR_SAT_WRITE_MAX	Maximum SAT Writing Score (from any Test)
UHIR_SAT_CRIT_MAX	Maximum SAT Critical Reading Score (from any Test)
UHIR_ACT_COMP	ACT Composite Score
UHIR_ACT_ENG	ACT Verbal Score (from Test with Highest Composite Score)
UHIR_ACT_MATH	ACT Math Score (from Test with Highest Composite Score)
UHIR_ACT_TEST_DT	ACT Test Date
UHIR_ACT_ENG_MAX	Maximum ACT Verbal Score (from any Test)
UHIR_ACT_MATH_MAX	Maximum ACT Math Score (from any Test)
UHIR_GRE_TEST_DT	GRE Test Date (from Test with Highest Composite Score)
UHIR_GRE_TOTAL	GRE Composite Score
UHIR_GRE_VERB	GRE Verbal Score (from Test with Highest Composite Score)
UHIR_GRE_QUAN	GRE Quantitative Score (from Test with Highest Composite Score)
UHIR_GRE_ANAL	GRE Analytical Score (from Test with Highest Composite Score)
UHIR_GRE_AW	GRE Analytical Writing Score (from Test with Highest Composite Score)
UHIR_GRE_TEST_DTV	GRE Test Date (from Test with Highest Verbal Score)
UHIR_GRE_VERB_MAX	Maximum GRE Verbal Score (from any Test)
UHIR_GRE_VERB_PMAX	Percentile of Maximum GRE Verbal Score (from any Test)
UHIR_GRE_TEST_DTQ	GRE Test Date (from Test with Highest Quantitative Score)
UHIR_GRE_QUAN_MAX	Maximum GRE Quantitative Score (from any Test)
UHIR_GRE_QUAN_PMAX	Percentile of Maximum GRE Quantitative Score (from any Test)
UHIR_GRE_TEST_DTA	GRE Test Date (from Test with Highest Analytical Score)
UHIR_GRE_ANAL_MAX	Maximum GRE Analytical Score (from any Test)
UHIR_GRE_ANAL_PMAX	Percentile of Maximum GRE Analytical Score (from any Test)
UHIR_GRE_TEST_DTAW	GRE Analytical Writing Maximum Test Date (from any Test)
UHIR_GRE_AW_MAX	Maximum GRE Analytical Writing (from any Test)
UHIR_GRE_AW_PMAX	Percentile of Maximum GRE Analytical Writing (from any Test)
UHS_GRE_RV_TEST_DT	GRE Revised Test Date (from Revised Test with Highest Composite Score)
UHS_GRE_RV_TOTAL	GRE Revised Total (from Revised Test with Highest Composite Score)
UHS_GRE_RV_VERB	GRE Revised Verbal Score (from Revised Test with Highest Composite Score)

UHS_GRE_RV_QUAN	GRE Revised Quantitative Score (from Revised Test with Highest Composite Score)
UHS_GRE_RV_AW	GRE Revised Analytical Wrting Score (from Revised Test with Highest Composite Score)
UHS_GRE_RV_TEST_DV	GRE Revised Verbal Test Date
UHS_GRE_RV_VERB_MX	Maximum GRE Revised Verbal Score (from any Test)
UHS_GRE_RV_VERB_PM	Percentile of Maximum GRE Revised Verbal Score (from any Test)
UHS_GRE_RV_TEST_DQ	GRE Revised Quantitative Test Date (from any Test)
UHS_GRE_RV_QUAN_MX	Maximum GRE Revised Quantitative Score (from any Test)
UHS_GRE_RV_QUAN_PM	Percentile of Maximum GRE Revised Quantitative Score (from any Test)
UHIR_MAX_GMATTOTL	Maximum GMAT Total Score
UHIR_PERC_GMATTOTL	Percentile of Maximum GMAT Total Score
UHIR_GMAT_TEST_DTT	GMAT Total Test Date
UHIR_MAX_GMATANL	Maximum GMAT Analytical Score
UHIR_PERC_GMATANL	Percentile of Maximum GMAT Analytical Score
UHIR_GMAT_TEST_DTA	GMAT Analytical Test Date
UHIR_MAX_GMATQNT	Maximum GMAT Quantitative Score
UHIR_PERC_GMATQNT	Percentile of Maximum GMAT Quantitative Score
UHIR_GMAT_TEST_DTQ	GMAT Quantitative Test Date
UHIR_MAX_GMATVRB	Maximum GMAT Verbal Score
UHIR_PERC_GMATVRB	Percentile of Maximum GMAT Verbal Score
UHIR_GMAT_TEST_DTV	GMAT Verbal Test Date
UHIR_LSAT_TOTAL	LSAT Total Score
UHIR_LSAT_TEST_DT	LSAT Test Date
UHIR_MAT	MAT Score
UHIR_MAT_TEST_DT	MAT Test Date
UHIR_OAT	OAT Score
UHIR_TOEFL_COMP	TOEFL Comprehensive
UHIR_TOEFL_TST_DTC	TOEFL Comp Test Date
UHIR_TOEFL_TOTAL	TOEFL Total
UHIR_TOEFL_TST_DTT	TOEFL Total Test Date
UHIR_ITOEFL_COMP	ITOEFL Comprehensive
UHIR_ITOEFL_TST_DT	ITOEFL Test Date
UHIR_PCAT_MATH	PCAT Math
UHIR_PCAT_VERB	PCAT Verbal
UHIR_PCAT_QUAN	PCAT Quantitative
UHIR_PCAT_READ	PCAT Reading
UHIR_PCAT_BIO	PCAT Biology
UHIR_PCAT_CHEM	PCAT Chemistry
UHIR_IELTS_OVERALL	IELTS Overall
UHIR_IELTS_LSTN	IELTS Listen
UHIR_IELTS_READ	IELTS Reading
UHIR_IELTS_SPEAK	IELTS Speaking
UHIR_IELTS_WRITE	IELTS Writing
UHIR_IELTS_TEST_DT	IELTS Test Dt
UHIR_LEVEL6_WVR_DT	Level 6 TOEFL Waiver Date
STRM	Term (same value as ADMIT_TERM)
UHIR_ACT_ELA	ACT English Language Arts
UHIR_ACT_STEM	ACT STEM Score
UHIR_ACT_WDDS	ACT Writing
UHIR_ACT_WDIA	ACTWriting Ideas and Analys

UHIR_ACT_WDO	ACT Write Domain Organization
UHIR_ACT_WDTC	ACT Write Domain Lang. Conver
UHIR_ACT_WRS	ACT Writing Subject Score
UHIR_SAT_MSS	SAT Math Section Score
UHIR_SAT_ERWS	SAT Read/Writing Sect Score
UHIR_SAT_MT	SAT Math Test Score
UHIR_SAT_RT	SAT Reading Test Score
UHIR_SAT_TOTAL	SAT Total Score
UHIR_NEW_SAT_TSTDT	SAT New Test Date
UHIR_SAT_MSS_MAX	SAT Math Section Score Max
UHIR_SAT_ERWS_MAX	SAT Reading Test Score
UHIR_SAT_AHSSC	SAT Hist/SocSci XTest Score
UHIR_SAT_ASC	SAT Science XTest Score
UHIR_SAT_CE	SAT Cmmnd Of Evidence Subscre
UHIR_SAT_EI	SAT Expressn of Ideas Subscre
UHIR_SAT_ESA	SAT Essay Analysis Subscore
UHIR_SAT_ESR	SAT Essay Reading Subscore
UHIR_SAT_ESW	SAT Essay Writing Subscore
UHIR_SAT_HA	SAT Heart Of Algebra Sbscre
UHIR_SAT_PAM	SAT Pspt to Adv Math Sbscre
UHIR_SAT_PSDA	SAT ProbSlv/Data Anly Sbscre
UHIR_SAT_RWC	SAT Rel Words In Ctxt Sscore
UHIR_SAT_SEC	SAT Stnd Engl Conv Subscore
UHIR_SAT_WLT	SAT Writing/Lang Test Score

Table: PS_ UHIR_ CRSE_ATTR

Record Name: **PS_UHIR_CRSE_ATTR**

Record Description: This reporting table record contains data related to the classes offered at one of the U of H institutions. This table currently contains class instructors and attributes.

Field Names	Field Descriptions
CRSE_ID (Key)	Course ID
CRSE_OFFER_NBR (Key)	Course Offering Nbr
STRM (Key)	The semester (term) for which the class data is associated.
SESSION_CODE (Key)	The session code (i.e. Regular or 8 weeks first)
CLASS_SECTION (Key)	The class section
INSTR_ASSIGN_SEQ (Key)	Class/Instructor Assign Seq #
CLASS_MTG_NBR (Key)	Class Meeting Pattern Nbr
CRSE_ATTR (Key)	Course Attribute
CRSE_ATTR_VALUE (Key)	Course Attribute Value
CRSE_ATTR_DESCR	Course Attribute Description
CRSE_ATTRVAL_DESCR	Course Attribute Value Descrip
INSTRUCTOR_ID	Instructor ID (There could be multiple instructors per class)
INSTR_NAME	Instructor
INSTR_ROLE	Instructor Role
INSTITUTION	Academic Institution
CATALOG_NBR	Catalog Nbr

Table: PS_ UHS_STDNT_LOC

Record Name: **PS_UHS_STDNT_LOC**

Record Description: This table shows the locations and/or on-line class indication for which a UH student is taking classes each semester. If a student is taking classes at more than one location, both will be checked. A student can be both on-line and at a location.

Field Names	Field Descriptions
EMPLID (Key)	Empl ID
INSTITUTION (Key)	Academic Institution
STRM (Key)	Term
LAST_NAME	Last Name
FIRST_NAME	First Name
MIDDLE_NAME	Middle Name
UHS_CAMPUS_LOC	Main Campus Location: Y=at least one class on-campus, N=no classes on-campus.
UHS_SUGAR_LAND	Sugar Land Campus: Y=at least one class at Sugarland, N=no classes at Sugarland.
UHS_CINCO_RANCH	Cinco Ranch Campus: Y=at least one class at Cinco Ranch, N=no classes at Cinco Ranch.
UHS_NORTHWEST_LOC	Northwest Campus: Y=at least one class at UH-Northwest, N=no classes at UH-Northwest.
UHS_OTHER_OFFCMPUS	Other Location Off Campsus: Y=at least one class at other off-campus location, No=no classes at other off-campus locations.
UHS_ONLINE_ONLY	Online Only: Y=Student has only on-line classes, N=not only on-line classes.
UHS_OLO_OUTOFTX	Online Only Out Of TX - **NO DATA AS OF DECEMBER, 2015**
UHS_NONE_ONLINE	No Online Class: Y=Student has no on-line classes, N=has some or only on-line classes.

Note: If the student has some on-line and some face-to-face classes, then both the UHS_ONLINE_ONLY and UHS_NONE_ONLINE fields will be N.

Table: PS_UHM_UGRD_COHORT - UHS UGRD Cohort Data

Record Name: **PS_UHM_UGRD_COHORT**

Record Description: This table identifies new undergraduate students to UH each semester. It identifies FTIC and Transfer students, including the degree-seeking and/or full-time degree seeking students.

Field Names	Field Descriptions
INSTITUTION	Academic Institution
SSR_COHORT_YR	Cohort - Identifies the calendar year that the student entered.
SSR_COHORT_TERM	Cohort Term - Identifies the specific term for the Cohort. The FTIC cohort will be the combined summer/fall FTIC population.
EMPLID	Empl ID
UHS_COHORT_FTIC	UHS Cohort FTIC - A Y means a part of the summer/fall FTIC cohort, N is not.
UHS_COHORT_FTIC_FA	UHS Cohort FTIC Fall only - the FTIC cohort student started in the fall semester.
UHS_COHORT_FTIC_SU	UHS COHORT FTIC from Summer - the FTIC cohort student started in the summer semester
UHS_COHORT_TRANSFE	UHS_Cohort_Transfer - A Y means the student is part of the new Transfer cohort.
UHS_COHORT_DEGSEEK	UHS Cohort Degree Seeking - the new student is degree-seeking
UHS_COHORT_FTMDEG	UHS Cohort full-time deg seek - A Y means the student is a new full-time, degree-seeking student.
COMMENTS	Comment about changes to a student status.
UHS_TIME_DONE	Completed Time - when data was loaded into the table.
OPRID2	User ID that ran the process to populate the table.

Table: PS_UHM_ADM_FTICPOP - UHM ADMISSION FTIC PRE-ENROLLMENT TABLE

Record Name: **PS_UHM_ADM_FTICPOP**

Record Description: This table populates for FTIC applicants who have been admitted to the University. It gives key information about them to help understand where they are with moving toward enrollment. At the moment only applies for Spring 2015 forward.

Field Names	Field Descriptions
INSTITUTION	Academic Institution
ADMIT_TERM	Admit Term
EMPLID	Empl ID
LAST_NAME	Last Name
FIRST_NAME	First Name
MIDDLE_NAME	Middle Name
SSR_COHORT_YR	Cohort-Identifies the calendar year that the student entered. e.g., 2015, 2014
SSR_COHORT_TERM	Cohort Term-Identifies the specific term for the Cohort. The FTIC cohort will be the combined summer/fall FTIC population. e.g., 1970, 1980 etc.
UHS_COHORT_FTIC	UHS Cohort FTIC - Y= part of the summer/fall FTIC cohort, N is not.
UHS_COHORT_FTIC_FA	UHS Cohort FTIC Fall only - Y= the FTIC cohort student started in the fall semester.
UHS_COHORT_FTIC_SU	UHS COHORT FTIC from Summer - Y=the FTIC cohort student started in the summer semester
UHS_COHORT_DEGSEEK	UHS Cohort Degree Seeking - Y=student is degree-seeking
UHS_COHORT_FTMDEG	UHS Cohort full-time deg seek - Y=the student is a new full-time, degree-seeking student.
ADMIT_TYPE	Admit Type
UHIR_NEW_EDESC_CB	IR CB New Ethnic Description
SEX	Gender
UHS_VETERAN_STATUS	Veteran Status = Active Duty, Spouse, Dependent, Veteran
UHIR_IN_HOUSING	In UH housing as of term census
UHS_1ST_GENERATION	1st Generation Status = Y if neither parent has a bachelor degree or higher.
PRIMARY_EFC	Primary Expected Family Contribution from financial aid application.
PELL_ELIGIBILITY	Federal PELL Eligibility = Y, N = not Pell Elgible, Blank=no Financial Aid data submitted
UHS_ADMIT_TERM_ENR	Enrolled in Admit_Term
UHS_ADM_ENRL_UNITS	Total Admit_Term_Units Enrolled
UHS_CUR_ACAD_LVL	Academic Level from Student Records side
UHS_APPL_GROUP1	Initial Application College 1
UHS_APPL_GRP_DESC1	Initial application coll 1 des
UHS_APPL_PLAN_1	Appl 1st coll 1st acad_plan
UHS_APPL_PLA_DESC1	Appl Coll 1 1st acad_plan desc
UHS_APPL_ORG_DESC1	Appl Coll 1 1st plan dept desc
UHS_APPL_GROUP2	Initial Application College 2
UHS_APPL_GRP_DESC2	Initial application coll 2 des
UHS_APPL_PLAN_2	Appl 2nd coll 1st acad_plan
UHS_APPL_PLA_DESC2	Appl Coll 2 1st acad_plan desc
UHS_APPL_ORG_DESC2	Appl Coll 2 1st plan dept desc
UHS_APPL_GROUP3	Initial Application College 3
UHS_APPL_GRP_DESC3	Initial application coll 3 des
UHS_APPL_PLAN_3	Appl 3rd coll 1st acad_plan
UHS_APPL_PLA_DESC3	Appl Coll 3 1st acad_plan desc

UHS_APPL_ORG_DESC3	Appl Coll 3 1st plan dept desc
UHS_ADMIT_GROUP1	Admitted College 1
UHS_ADMT_GRP_DESC1	First admitted college descr
UHS_ADMIT_PLAN_1	1st Admit Coll first acad_plan
UHS_ADMT_PLA_DESC1	1st admt coll 1st plan descr
UHS_ADMT_ORG_DESC1	1st admt coll 1st plan dep des
UHS_ADMIT_PLAN_12	1st admit coll 2nd acad_plan
UHS_ADMT_PLA_DES12	1st admt coll 2nd plan descr
UHS_ADMT_ORG_DES12	1st Admt Coll 2nd plan Dep des
UHS_ADMIT_GROUP2	Admitted College 2
UHS_ADMT_GRP_DESC2	Second admitted college descr
UHS_ADMIT_PLAN_2	2nd admit coll first acad_plan
UHS_ADMT_PLA_DESC2	2nd admt coll 1st plan descr
UHS_ADMT_ORG_DESC2	2nd admt coll 1st plan dep des
UHS_ADMIT_PLAN_22	2nd admit coll 2nd acad_plan
UHS_ADMT_PLA_DES22	2nd admit coll 2nd plan descr
UHS_ADMT_ORG_DES22	2nd Admt Coll 2nd plan Dep des
UH_HONORS_FLAG	In the Honors
UHS_UH4_SEL_DESCR	UH4 Selection Decsr - will show "not eligible", "yes", "no", and "undecided". If the student does not have an entry in UHS_UH4_STU_SEL record, then this field is blank.
UHS_UH4_SEL_STA_DS	Selection Status for UH in 4 - "Confirmed", "Denied", "Pending", or "Requirement not met". If the student does not have an entry in UHS_UH4_STU_SEL record, then this field is blank.
UHS_TSI_HOLD	TSI Hold Servic Indicator Code
UHS_UP_COLLECTION	UHS_UP_COLLECTION_SRVC_IND - f the student has currently active "B11" service indicator in SRVC_IND_DATA table, then this UHS_UP_COLLECTION field is "Y" otherwise this UHS_UP_COLLECTION field is "N".
UHS_FINA_DELINQUEN	UHS FInance Delinquency SRVC - If the student has currently active "B12" service indicator in SRVC_IND_DATA table, then this UHS_FINA_DELINQUEN field is "Y" otherwise this UHS_FINA_DELINQUEN field is "N".
UHS_BAD_DEBIT	UHS_BAD_DEBIT_SRVC_IND - If the student has currently active "B13" service indicator in SRVC_IND_DATA table, then this UHS_BAD_DEBIT field is "Y" otherwise this UHS_BAD_DEBIT field is "N".
UHS_ACCOUNT_BALANC	Account Balance - This is the summation of account_balance from ACCOUNT_SF for the student, from all tems and accounts.
UHS_SERVICE_IMPACT	Different codes indicate if a student has a hold that prevents all enrollment activity, or if they can drop but not add, etc.
UHIR_SAT_CRIT_MAX	Maximum SAT Critical Reading
UHIR_SAT_MATH_MAX	Maximum SAT Math
UHIR_SAT_WRITE_MAX	Maximum SAT Writing
UHIR_SAT_COMP	SAT Composite Max Score
UHIR_ACT_ENG_MAX	Maximum ACT Verbal
UHIR_ACT_MATH_MAX	Maximum ACT Math
UHIR_ACT_COMP	ACT Comprehensive Score for single date
UHIR_HS_NAME	High School Name
UHIR_HS_STATE	High School State
UHS_HS_ZIP	High School Zip code
UHS_HS_COUNTY	County
UHS_HS_CITY	City
UHIR_HS_PERCENTILE	High School Percentile

UHIR_HS_GRAD_DT	High School Graduation Date
UHS_TSI_MATH_STAT	TSI Math Status
UHS_TSI_READ_STAT	TSI Read Status
UHS_TSI_WRITE_STAT	TSI Write Status
UHS_TSI_STATUS	TSI Component Status
UHS_ORIENTATION_DT	Orientation Meeting Date
UHS_ORIENT_EVENTID	Event ID for Orientation
UHS_ORIEN_EVNT_STA	Orientation Event Meeting Attendee Status
UHS_UH_EMAIL_ADDR	UH Email Addresses
UHS_DES_EMAIL_ADDR	Designated Email Addresses
UHS_OTH_EMAIL_ADDR	Other Email Address
PHONE	Telephone
UHS_CELL_PHONE	Cell Phone
BIRTHDATE	Date of Birth
UHIR_MAIL_ADDRESS1	Mail Address 1
UHIR_MAIL_ADDRESS2	Mail Address 2
UHIR_MAIL_ADDRESS3	Mail Address 3
UHIR_MAIL_ADDRESS4	Mail Address 4
UHIR_MAIL_CITY	Mail City
UHIR_MAIL_STATE	Mail State
UHIR_MAIL_POSTAL	Mail Postal Zip code
UHIR_MAIL_COUNTY	Mailing Address County
UHIR_MAIL_COUNTRY	Mailing Address Country
COMMENTS	Comment about when date was loaded and the Process Instance number
UHS_TIME_DONE	Completed Time
OPRID2	User ID

Table: PS_UHM_FTIC_COHDAT - UHM FTIC Cohort Post-Table

Record Name: **PS_UHM_FTIC_COHDAT**

Record Description: This table populates data on each enrolled FTIC cohort. It provides status information each semester. .

Field Names	Field Descriptions
INSTITUTION	Academic Institution
SSR_COHORT_YR	Cohort-Identifies the calendar year that the student entered. e.g., 2015, 2014
SSR_COHORT_TERM	Cohort Term-Identifies the term that the student entered. The fall cohort terms will include the combined summer/fall FTIC population. e.g., 1970, 1940
EMPLID	Empl ID
STRM	Term
LAST_NAME	Last Name
FIRST_NAME	First Name
MIDDLE_NAME	Middle Name
UHS_COHORT_FTIC	UHS Cohort FTIC - Y= part of the summer/fall FTIC cohort, N is not.
UHS_COHORT_FTIC_FA	UHS Cohort FTIC Fall only - Y= the FTIC cohort student started in the fall semester.
UHS_COHORT_FTIC_SU	UHS COHORT FTIC from Summer - Y=the FTIC cohort student started in the summer semester
UHS_COHORT_DEGSEEK	UHS Cohort Degree Seeking - Y=student is degree-seeking
UHS_COHORT_FTMDEG	UHS Cohort full-time deg seek - Y=the student is a full-time, degree-seeking student.
UHIR_NEW_EDESC_CB	IR CB New Ethnic Description
SEX	Gender
UHS_VETERAN_STATUS	Veteran Status = Active Duty, Spouse, Dependent, Veteran
UHIR_IN_HOUSING	In housing as of term census
UHS_1ST_GENERATION	1st Generation Status = Y if neither parent has a bachelor degree or higher.
PRIMARY_EFC	Primary Expected Family Contribution from financial aid application.
PELL_ELIGIBILITY	Federal PELL Eligibility = Y, N = not Pell Eligible, Blank=no Financial Aid data submitted
ENROLLED	Enrolled Y/N. If current or future term, current enrollment. If past term, EOS enrollment.
UHS_PRE_ENRL_STRM	Most recent previous term of enrollment
UHS_PRE_ENRL_UNITS	Previous enrl term: enrolled units
UHS_PRE_PASS_UNITS	Previous enrl term: passed units
UHS_PRE_CUM_GPA	Previous enrl term: cum GPA
UHS_PRE_TERM_GPA	Previous enrl term: GPA
UHS_PRE_DEAN_LIST	Previous enrl term: Dean's List
UHPRE_ACAD_STNDDES	Previous enrl term: academic standing
UHS_PRE_TOT_CUMHRS	Previous enrl term: total cum hours
UHS_PRE_GROUP1	Previous enrl term: College 1
UHS_PRE_GRP_DESC1	Previous enrl term: College 1 Descr
UHS_PRE_PLAN_1A	Previous enrl term: Plan 1A
UHS_PRE_PLAN_DES1A	Previous enrl term: Plan 1A Descr
UHS_PRE_ORG_DESC1	Previous enrl term: Plan 1A Dept
UNT_TAKEN_PRGRSS	Units Enrolled for Progress
UNT_PASSD_PRGRSS	Units Passed for Progress
UHS_CUM_GPA	Cumulative GPA
UHS_CUR_GPA	Current GPA

UHIR_DEANS_LIST	Dean's List Y/N
UHIR_ACAD_STNDDESC	Current Acad Standing Desc
UHS_TOT_CUMULATIVE	Total Cumulative Units
UHS_TOT_TAKEN_PRGR	Total Units Taken for Progress
UHS_TOT_PASSD_PRGR	Total Units Passed for Progress
UHS_PLAN_UNITS_REQ	DPR Plan Units Required
UHS_PLAN_UNIT_USED	DPR Plan Units Completed
UHS_DEG_PRCNT_COMP	DPR Plan Percent Complete
UHS_DRP_RUN_DT	DPR Last Run Date
UHS_GPA_CURR_PLAN	GPA in current DPR plan
UHS_DEGR_CHK_STAT	Degree Checkout Status
UHS_DEGRE_CHKOUTDT	Degree Checkout Effect Date
EXP_GRAD_TERM	Expected Graduation Term
UHS_DEGREE_1	UH DEGREE 1
UH_DEGR1_COMP_TERM	Degree 1 Completion Term
UHS_DEGREE_2	UH DEGREE 2
UH_DEGR2_COMP_TERM	Degree 2 Completion Term
UHIR_ACAD_LEVEL	Acad Level Description
UHS_1ST_GROUP1	1st Enrolled College 1
UHS_1ST_GRP_DESC1	1st Enrolled College1 Descr
UHS_1ST_PLAN_1A	1st Enrolled Plan 1A
UHS_1ST_PLAN_DES1A	1st Enrolled Plan 1A descr
UHS_1ST_ORG_DES1A	1st Enrolled Plan 1A Dept Descr
UHS_1ST_PLAN_1B	1st Enrolled Plan 1B
UHS_1ST_PLAN_DES1B	1st Enrolled Plan 1B descr
UHS_1ST_ORG_DES1B	1st Enrolled Plan 1B Dept Descr
UHS_1ST_GROUP2	1st Enrolled College 2
UHS_1ST_GRP_DESC2	1st Enrolled College 2 Descr
UHS_1ST_PLAN_2A	1st Enrolled Plan 2A
UHS_1ST_PLAN_DES2A	1st Enrolled Plan 2A descr
UHS_1ST_ORG_DES2A	1st Enrolled Plan 2A Dept Descr
UHS_1ST_PLAN_2B	1st Enrolled Plan 2B
UHS_1ST_PLAN_DES2B	1st Enrolled Plan 2B descr
UHS_1ST_ORG_DES2B	1st Enrolled Plan 2B Dept Desc
UHS_CUR_GROUP1	Current College 1
UHS_CUR_GRP_DESC1	Current College1 Descr
UHS_CUR_PLAN_1A	Current Plan 1A
UHS_CUR_PLAN_DES1A	Current Plan 1A descr
UHS_CUR_ORG_DES1A	Current Plan 1A Dept Descr
UHS_CUR_PLAN_1B	Current Plan 1B
UHS_CUR_PLAN_DES1B	Current Plan 1B descr
UHS_CUR_ORG_DES1B	Current Plan 1B Dept Descr
UHS_CUR_GROUP2	Curernt College 2
UHS_CUR_GRP_DESC2	Current College 2 Descr
UHS_CUR_PLAN_2A	Current Plan 2A
UHS_CUR_PLAN_DES2A	Current Plan 2A descr
UHS_CUR_ORG_DES2A	Current Plan 2A Dept Descr
UHS_CUR_PLAN_2B	Current Plan 2B
UHS_CUR_PLAN_DES2B	Current Plan 2B descr
UHS_CUR_ORG_DES2B	Current Plan 2B Dept Descr
UHS_CUR_MINOR_1A	Minor Plan 1A

UHS_CUR_MINORDES1A	Minor Plan 1A Descr
UHS_CUR_MINOR_2A	Minor Plan 2A
UHS_CUR_MINORDES2A	Minor Plan 2A Descr
UH_HONORS_FLAG	Honors College Y/N
UHS_UH4_SELECTION	UHin4 Y/N
UHS_TSI_HOLD	TSI Hold Service Indicator Y/N
UHS_UP_COLLECTION	B11 Hold Service Indicator Y/N (Up Collections)
UHS_FINA_DELINQUEN	B12 Hold Service Indicator Y/N (Financial Delinquency)
UHS_BAD_DEBIT	B13 Hold Service Indicator Y/N (Bad Debt)
UHS_ACCOUNT_BALANC	Total account balance from all accounts and terms
UHS_SERVICE_IMPACT	Service impact codes, e.g., AENR
UHIR_SAT_COMP	SAT Composite Score
UHIR_ACT_COMP	ACT Composite Score
UHIR_HS_NAME	High School Name
UHIR_HS_GRAD_DT	High School Graduation Date
UHS_TSI_STATUS	TSI Component Status
UHS_UH_EMAIL_ADDR	Email Addresses
UHS_DES_EMAIL_ADDR	Email Addresses
UHS_OTH_EMAIL_ADDR	Email Addresses
PHONE	Telephone
UHS_CELL_PHONE	Cell Phone
BIRTHDATE	Date of Birth
UHIR_MAIL_ADDRESS1	Mail Address 1
UHIR_MAIL_ADDRESS2	Mail Address 2
UHIR_MAIL_ADDRESS3	Mail Address 3
UHIR_MAIL_ADDRESS4	Mail Address 4
UHIR_MAIL_CITY	Mail City
UHIR_MAIL_STATE	Mail State
UHIR_MAIL_POSTAL	Mail Postal Zipcode
UHIR_MAIL_COUNTY	Mailing Address County
UHIR_MAIL_COUNTRY	Mailing Address Country
UHS_DECEASED_STDNT	Student is Deceased Y/N
COMMENTS	Comment
UHS_TIME_DONE	Completed Time
OPRID2	User ID

Housing Tables – Data Dictionary

Record Name: **PS_UHIR_HOUSING**

Record Description: This record contains information from the UH Housing data system which indicates that a person either has a housing application in process or is booked into a room in UH Housing. The data is pulled out of the Housing data system and moved into myUH nightly.

Field Names	Field Description	Details
Institution (Key)	Academic Institution	Should always be 00730 for UH Managed Housing
STRM (Key)	Term	
Emplid (Key)	Student PeopleSoft ID	Student PeopleSoft id
UHIR_APP_ID (Key)	Housing Application ID Number	Housing Application ID
UHIR_APP_STAT_CD	Housing Appl Status Code	Indicates where in the applying for housing process a student is.
UHIR_APP_STATUS	Appl. Status Description	a student's status with the housing application.
UHIR_APP_FEE_PAID	Appl Fee paid flag-Y=paid	Y = student has taken an action to pay the application fee. N = not an indicator of whether fee was paid or not.
UHIR_HOUSE_PREPAY	Prepayment Paid Flag-Y=paid	Y = student has taken an action to pay the pre-payment (deposit). N = not an indicator of whether fee was paid or not.
UHIR_HOUSE_WAIVER	Waiver Flag-Y=UH fee waiver	Y = student has an application fee waiver for UH – therefore App Fee and Deposit Paid fields should reflect as "N" in most circumstances. N = student does not have an application fee waiver for UH.
UHIR_APP_TYPE	Applt: 0=new appl; 2=retnr	Indicates whether the person in housing is returning or new applicant. 0 = regular / new application 2 = returner application. Returner applications will not be typical after priority room selection which typically ends each year on February 28.
UHIR_MOVEIN_DT	Room Move-in date	Date the student is expected to move into their booked room. This field will be empty if the student has not booked a room or if their status is "unbookedsigned." Prior to a term, this date will be the same until students are allowed tomove in to UH housing for the term.
UHIR_APPCANCEL_DT	Room bking or appl cancel date	The date the booking for a room is effectively cancelled or when an application has been cancelled before a booking is complete. For a current booking this field will be empty. It will also be empty if a student has not made it all the way through the booking process.
UHIR_BOOKING_DT	The date booking complete	The date a booking was complete
UHIR_COMNTY_CD	Housing Community Code	Number that indicates the UH Housing community, e.g., Cougar Village 1, Calhoun Lofts, etc.
UHIR_COMNTY_DESC	Community Description/name	Name for UH Housing Community
UHIR_BLDG_CODE	Bldg Code in Housing	Code of building within a UH Housing Community

	Community	
UHIR_BLDG_DESC	Bldg descr:ex. Clhn Lofts-West	Description for building in a UH Housing Community, e.g., Calhoun Lofts-West
UHIR_ASSGN_BEDID	Students assigned Rm/Bed ID	Bed ID assigned to student
UHIR_BEDID_DESC	Room/Bed Assignment Descr	Describes the type of room assigned to student
MAILBOX_ID	Mailbox number	Mailbox number
STREET1	Street Address UH Housing	Street Address for student's UH Housing residence
City	City for UH Housing	City for student's UH Housing residence
ZIP CODE	Zip code for UH Housing	Zip Code for student's UH Housing residence
STATE	State for UH Housing	State for student's UH Housing residence
UHIR_PERTYPE_APPL	Person=Applt has applied to UH	Indicates for Housing that a person has applied to UH
UHIR_PERTYPE_ENRL	Person=elgible to enroll at UH	Indicates for Housing that a person is eligible to enroll at UH
UHIR_PERTYPE_GRAP	Person=GR/SP Applt appld to UH	Indicates for Housing that a person has applied for graduate/professional school at UH
UHIR_PERTYPE_LCC	Person=person in LCC program	Indicates for Housing that a person is in the LCC program
UHIR_PERTYPE_NSTU	Person=Non Student in Housing	Indicates a person living in UH Housing is not a student.

UHIR_Housing Reporting Table Code Book

UHIR_APP_STAT_CD	UHIR_APP_STATUS - Description	
1	App Started	Student has logged into housing application system and started any one of four initial steps.
2	Paid	Student has paid app fee and prepayment but has not progressed to digital signature.
3	Booked	Student has a booked space but no signed agreement. Typically converts to BookedSigned prior to move in.
4	Unbooked	A previously Booked agreement was canceled and has yet to be rebooked. Will typically convert to Booked or BookedSigned
5	Signed	Student has successfully signed agreement, electronically or paper hardcopy.
6	BookedSentToSignix	Student is booked and their service agreement has been submitted to third party vendor to secure electronic signature. Will typically convert to BookedSigned prior to move in.
7	UnbookedSentToSignix	A previously BookedSenttoSignix booking was cancelled before electronic signature could be obtained. Rare occurrence and will typically convert to BookedSigned prior to move in.
8	SenttoSignix	Student has paid application fee and prepayment and their service agreement has been submitted to third party vendor to secure electronic signature.
9	BookedSigned	Student has been successfully assigned to a space.
10	UnbookedSigned	Student has had their assignment cancelled and it has not yet been rebooked. This only occurs upon administrative action. Student has signed an agreement and previously booked a space but booking has been canceled. Will typically convert to BookedSigned prior to move in.
11	Canceled	Either the student or administrator has cancelled the application
12	Received	No longer relevant to current business practice. No plans to re-implement but remains a possibility as it exists in database. If seen in tables then it is in error and would occur prior to payment.
13	Accepted	No longer relevant to current business practice. No plans to re-implement but remains a possibility as it exists in database. If seen in tables then it is in error and would occur prior to payment.

UHIR_COMNTY_CD	UHIR_COMNTY_DESC
1	Calhoun Lofts
2	Cougar Place Replacement
3	Cougar Village
4	Moody Tower
5	Quadrangle
7	Bayou Oaks

UHIR_BLDG_CODE	UHIR_BLDG_DESC
BO	Bayou Oaks
CLC	Calhoun Lofts -Central
CLE	Calhoun Lofts - East
CLN	Calhoun Lofts - North
CLS	Calhoun Lofts - South
CLW	Calhoun Lofts - West

CPR	Cougar Place Replacement
CV01	Cougar Village 1
CV02	Cougar Village 2
NT	Moody Tower - North Tower
QB	Quadrangle - Bates Hall
QL	Quadrangle - Law Hall
QO	Quadrangle - Oberholtzer Hall
QS	Quadrangle - Settegast Hall
QT	Quadrangle - Taub Hall
ST	Moody Tower - South Tower

UHIR_APP_FEE_PAID
Y = student has taken an action to pay the application fee.
N = not an indicator of whether fee was paid or not.

UHIR_HOUSE_PREPAY
Y = student has taken an action to pay the pre-payment (deposit).
N = not an indicator of whether fee was paid or not.

UHIR_HOUSE_WAIVER
Y = student has an application fee waiver for UH – therefore App Fee and Deposit Paid fields should reflect as "N" in most circumstances.
N = student does not have an application fee waiver for UH.

UHIR_APP_TYPE
0 = regular / new application
2 = returner application. Returner applications will not be typical after priority room selection which typically ends each year in February

UHIR_PERTYPE_APPL
Y = Applicant,indicates for Housing that a person has applied to UH
N = No such indication to Housing

UHIR_PERTYPE_ENRL
Y=Enrolled, indicates for Housing that a person is eligible to enroll at UH
N = No such indication to Housing

UHIR_PERTYPE_GRAP
Y=Indicates for Housing that a person has applied for graduate/professional school at UH
N = No such indication to Housing

UHIR_PERTYPE_LCC
Y=Indicates for Housing that a person is in the LCC program
N = No such indication to Housing

UHIR_PERTYPE_NSTU
Y=Non Student,indicates a person living in UH Housing is not a student.
N = No such indication to Housing

CB Tables – Data Dictionaries

UHCB_001 – Students Report

Field Names	Field Descriptions
EMPLID	EmplID
INSTITUTION	Academic Institution
STRM	Term
ACAD_CAREER	Academic Career
ACAD_PLAN	Academic Plan
UHCB_FLEX_ENTRY	CB Flexible Entry
UHCB_FUNDING_CODE	CB Funding Code
UHCB_SEMESTER	CB Semester
NATIONAL_ID	National ID
REQ_TERM	Requirement Term
UHCB_RECORD_CODE	CB Record Code
FICE_CD	FICE Code
SEX	Gender
UHCB_CLASIFICATION	CB Classification
UHCB_BIRTHDATE	UHCB Birthdate YYYYMMDD
UHCB_TUIT_STATUS	CB Tuition Status
UHCB_RESIDENCY	CB Residency
UHCB_ORIG_RSDNCE	CB Original Residence
UHCB_TRANSFER_TIME	CB Transfer First Time in Coll
UHCB_001_10A	SCH On campus
UHCB_001_10B	SCH Off campus
UHCB_001_11	Doctoral hours funded
UHCB_ETHNIC_ORIGIN	Ethnic Origin
UHCB_CALENDAR_YEAR	CB Calendar Year
UHCB_001_15	Institutional state fund
UHCB_REMOTE_CAMPUS	Remote
UHCB_MAJOR_CONCENT	CB Major Concentration
UHCB_TUIT_EXCEMPT	CB Tuition Exempt
UHCB_001_21	Collegiate not funded
UHCB_001_22	Development not funded
UHCB_001_23	Institutional not funded
UHCB_001_24	Collegiate state funded
UHCB_001_25	Developmental state funded
UHCB_001_26	Ugrad Degree Program
UHCB_FUND_LIMIT	CB UGRD Fund Limit
LAST_NAME	Last Name
FIRST_NAME	First Name
MIDDLE_NAME	Middle Name
UHCB_001_31	Dual Credit
UHCB_TEACH_ED_PROG	Teacher Education Program
UHCB_UPDATE_CODE	CB Update Code

UHCB_001_33_NONDGR	Non Degree Student
UHCB_NONDISCLOSURE	UHCB Non Disclosure
UHCB_001_35	Graduate SCH Sr 12hrs Graduatn
UHCB_001_36	CEEB High School Code
UHCB_001_37	PEIMS Student High School ID
UHS_CB_NEW_ETHNC_O	New CB Ethnic Origin
UHS_CB_WHITE_RACE	CB White
UHS_CB_BLACK_RACE	CB Black/African American
UHS_CB_ASIAN_RACE	CB Asian
UHS_CB_AMI_RACE	CB Amer Indian/Alaska Native
UHS_CB_INTNL_RACE	CB Race
UHS_CB_UNKNWN_RACE	CB Race
UHS_CB_PAC_RACE	CB NativeHawaiian/Pacificisland
PROCESS_DTTM	Process Date/Time
MANUAL_UPD_DTTM	Manual Update Date/Time Stamp
LAST_UPDT_OPRID	Last Updated User Name
UHS_PROCESS_NAME	Process Name

UHCB_003 – Course Inventory

Record Name: **UHCB_003**

Record Description: This table contains data reported on the CMB003 report .

Field Names	Field Descriptions
INSTITUTION	Academic Institution
ACAD_YEAR	Academic Year
SUBJECT	Subject Area
CATALOG_NBR	Catalog Nbr
CRSE_ID	Course ID
UHCB_RECORD_CODE	CB Record Code
FICE_CD	FICE Code
UNITS_MAXIMUM	Maximum Units
UHCB_LEVEL	Academic Level
CIP_CODE	CIP Code
UHCB_MULT_CRS_LIST	Multiple Course Listing
TITLE	Title
COURSE_TITLE_LONG	Long Course Title
UHCB_LAB_HRS	Lab Hours
UHCB_LEC_HRS	Lecture Hours
UHCB_ADMIN_UNIT_CD	Admin Unit Code
UHCB_UPDATE_CODE	CB Update Code
UHCB003_SUBMISSION	UHCB Submission Type
UHS_CB_RESBMT_CORR	Resubmit Corrections

UHCB_004 – Class Report

Record Name: **UHCB_004**

Record Description: This table contains data reported on the CMB004 report .

Field Names	Field Descriptions
INSTITUTION	Academic Institution
STRM	Term
CLASS_NBR	Class Nbr
NATIONAL_ID	National ID
UHCB_RECORD_CODE	CB Record Code
FICE_CD	FICE Code
SUBJECT	Subject Area
CATALOG_NBR	Catalog Nbr
CLASS_SECTION	Class Section
ACAD_GROUP	Academic Group
UHCB_INSTRUCT_TYPE	CB Instruction Type
UNITS_MAXIMUM	Maximum Units
UHCB_LOCATION	Location
UHCB_OTHER_FICE_CD	CB Other Fice Code
UHCB_COMPOSITE_CLS	CB Composite Class
UHCB_TENURE	CB Tenure
UHCB_OFF_CAMP_ZIP	CB Zip
UHCB_RESPONSE_FACT	CB Responsibility Factor
UHCB_004_13A	Enrollment 004, 13A
UHCB_004_13B	Enrollment 004, 13B
UHCB_004_13C	Enrollment 004, 13C
UHCB_004_13D	Enrollment 004, 13D
UHCB_004_13E	Enrollment 004, 13E
UHCB_SEMESTER	CB Semester
UHCB_CALENDAR_YEAR	CB Calendar Year
UHCB_004_16	Enrollment 004, 16
UHCB_004_17	Enrollment 004, 17
UHCB_004_18	Enrollment 004, 18
UHCB_004_19	Enrollment 004, 19
UHCB_INSTRUCT_MODE	CB Instruction Mode
UHCB_INTER_INST	CB Inter Institutional
UHCB_UPDATE_CODE	CB Update Code
UHCB_TEACH_LOAD_CR	CB Teaching Load Credit
PROCESS_DTTM	Process Date/Time
MANUAL_UPD_DTTM	Manual Update Date/Time Stamp
LAST_UPDT_OPRID	Last Updated User name
UHS_PROCESS_NAME	Process Name

UHCB_008 – Faculty Report

Record Name: **UHCB_008**

Record Description: This table contains data reported on the CMB008 report .

Field Names	Field Descriptions
EMPLID	EmplID
INSTITUTION	Academic Institution
STRM	Term
FICE_CD	FICE Code
NATIONAL_ID	National ID
LAST_NAME	Last Name
FIRST_NAME	First Name
MIDDLE_NAME	Middle Name
UHCB_TENURE	CB Tenure
SEX	Gender
UHCB_ADMIN_UNIT_CD	Admin Unit Code
BIRTH_DATE	
UHCB_APPOINT_01	CB Appointment 01 %
UHCB_APPOINT_02	CB Appointment 02 %
UHCB_APPOINT_11	CB Appointment 11 %
UHCB_APPOINT_12	CB Appointment 12 %
UHCB_APPOINT_13	CB Appointment 13 %
UHCB_SALARY_ST_APR	Salary - State Appropriations
UHCB_SALARY_DES	Salary - Designated
UHCB_SALARY_RES	Salary - Restricted
UHCB_SALARY_AUX	Salary - Auxilliary
UHCB_SALARY_OL	Salary - Overload
UHCB_FE_APPNT_PER	Flex Entry Appointment Percent
UHCB_FE_SALARY	Flex Entry Salary
UHCB_FLEX_ONLY	Flex Entry Only
UHCB_COMPLIANCE	CB Compliance
UHCB_NEW_HIRE	New Hire
UHCB_SEMESTER	CB Semester
UHCB_CALENDAR_YEAR	CB Calendar Year
UHCB_UPDATE_CODE	CB Update Code
UHCB_RECORD_CODE	CB Record Code
UHCB_RANK	CB Rank
UHCB_ETHNIC_GROUP	CB Ethnic Group
JOBCODE	Job Code
ACTION	Action
UHS_CB_NEW_ETHNIC_O	New CB Ethnic Origin
UHS_CB_WHITE_RACE	CB Race – White
UHS_CB_BLACK_RACE	CB Race – Black
UHS_CB_ASIAN_RACE	CB Race – Asian
UHS_CB_AMI_RACE	CB Race – Amer. Indian/Alaska Native
UHS_CB_INTNL_RACE	CB Race – International
UHS_CB_UNKNWN_RACE	CB Race – Unknown/Unreported
UHS_CB_PAC_RACE	CB Race – Native Hawaiian/Pacific Island
PROCESS_DTTM	Process Date/Time
MANUAL_UPD_DTTM	Manual Update Date/Time Stamp

LAST_UPDT_OPRID	Last Updated User Name
UHS_PROCESS_NAME	Process Name
UHS_CB_IPEDS_ETHNC	CB IPEDS Ethnicity
UHS_CB_IPEDS_EDESC	CB IPEDS Ethnic Description
SUBJECT	Subject Area
CATALOG_NBR	Catalog Number
UHCB_CLASS_SECTION	Class Section
UHCB_COMPOSITE_CLS	CB Composite Class
UHCB_RESPONSE_FACT	CB Responsibility Factor
UHCB_TEACH_LOAD_CR	CB Teaching Load Credit
UHCB_FLEX_ENTRY	CB Flexible Entry
CLASS_NBR	Class Number
UHS_CB_SEQ_NBR	CB Sequence Number

UHCB_00B

Record Name: **UHCB_00B**

Record Description: This table contains data reported on the CMB00B report .

Field Names	Field Descriptions
INSTITUTION	Academic Institution
STRM	Term
EMPLID	EmplID
NATIONAL_ID	National ID
FICE_CD	FICE Code
LAST_NAME	Last Name
FIRST_NAME	First Name
MIDDLE_NAME	Middle Name
SEX	Gender
BIRTH_DATE	Birthdate
ETHNIC_GROUP	Ethnic Group
UHCB_APPL_LEVL	Application Level
UHCB_RESIDENCY	CB Residency
UHCB_ENTRNG_STATUS	Entering Status
UHCB_ACCEPT_STATUS	Acceptance Status
UHCB_FTH_HGH_EDUC	Father's Highest Educ. Level
UHCB_MTH_HGH_EDUC	Mother's Highest Educ. Level
UHCB_SNGL_PAR_HOME	Single Parent Home
UHCB_FAM_GROSS_INC	Family's Gross Income
UHCB_LANGUAGE_FL	Language Fluency
UHCB_FAM_OBL	CB Family Obligations
UHCB_NUM_PERS_HHLD	Number of Persons in Household
UHCB_TERM_SOUGHT	Term Sought
UHCB_ACADEMIC_YEAR	CB Academic Year
UHS_CB_ACT_CMP_SCR	ACT Composite Score
UHS_CB_SAT_CMP_SCR	SAT Composite Score
UHCB_RECORD_CODE	CB Record Code

CHAPTER 2 – Logging In

Step 1: Log onto PC

- Select: "Switch User"
- Enter: your myUH User ID and Domain (i.e. Cougarnet) password and hit Ok
  - For alternate PC login: select option "This Computer", enter userid as **Computer#\PSTrain** (the computer number is found at the top of the monitor), and enter password as **Iamn2PS!**

Step 2: Navigate to the list of PeopleSoft databases (**Note:** You should save this link in your favorites.)

- You can navigate to the list by either typing in the url in your internet browser, or by clicking on, or copy/pasting the link provided in this document below
  - Link: http://www.uh.edu/infotech/php/template.php?nonsvc_id=536

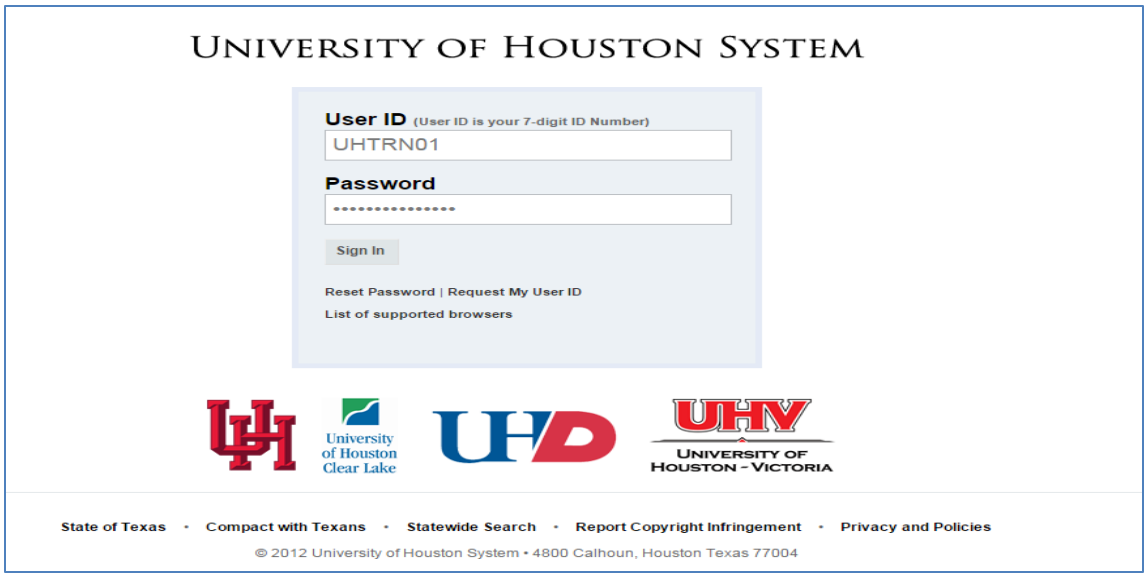
Step 3: Navigate to the Self Service Section and click on the database link given by the instructor.

- **PATRN9** https://dev.my.uh.edu:8328/
- **SA9TRN** https://dev.my.uh.edu:8329/psp/sa9trn/?cmd=login
- **SA9SBX** https://dev.my.uh.edu:8321/psp/sa9sbx/?cmd=login
- **SA9SBX2** https://dev.my.uh.edu:8325/psp/sa9sbx2/?cmd=login

Step 4: Log into the database training will be conducted (*will be provided by the instr uctor*)

- Training User IDs and Passwords:

User ID	Password	User ID	Password	User ID	Password
UHTRN01	4SUMMER2016@	UHTRN11	4SUMMER2016@	UHTRN21	4SUMMER2016@
UHTRN02	4SUMMER2016@	UHTRN12	4SUMMER2016@	UHTRN22	4SUMMER2016@
UHTRN03	4SUMMER2016@	UHTRN13	4SUMMER2016@	UHTRN23	4SUMMER2016@
UHTRN04	4SUMMER2016@	UHTRN14	4SUMMER2016@	UHTRN24	4SUMMER2016@
UHTRN05	4SUMMER2016@	UHTRN15	4SUMMER2016@	UHTRN25	4SUMMER2016@
UHTRN06	4SUMMER2016@	UHTRN16	4SUMMER2016@	UHTRN26	
UHTRN07	4SUMMER2016@	UHTRN17	4SUMMER2016@	UHTRN27	
UHTRN08	4SUMMER2016@	UHTRN18	4SUMMER2016@	UHTRN28	
UHTRN09	4SUMMER2016@	UHTRN19	4SUMMER2016@	UHTRN29	
UHTRN10	4SUMMER2016@	UHTRN20	4SUMMER2016@	UHTRN30	



CHAPTER 3 – PeopleSoft Query Elementals

Overview

In this chapter, you will learn the functionality highlights of Query Viewer and Query Manager, PeopleSoft Query output formats, how to perform query searches, effective dating, and how to navigate the PeopleSoft Query tool.

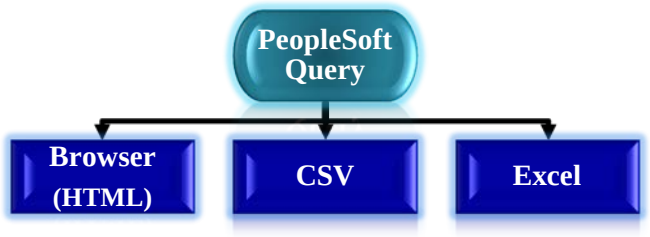
Query Viewer vs Query Manager

Within its toolset, PeopleSoft offers the components **Query Viewer** and **Query Manager** for accessing queries. Query Viewer is a read-only version of Query Manager. A limited population of users is granted security to Query Manager. Below are highlights of the functionality provided by these components:

Functionality	Available in Query Viewer	Available in Query Manager
✓ Search for a query (basic and advanced searches)	Yes	Yes
✓ Run/Execute queries	Yes	Yes
✓ Download query results	Yes	Yes
✓ Schedule a query	Yes	Yes
✓ Create/Edit queries	**No**	Yes

Query Reporting Formats

PeopleSoft Query automatically outputs to several reporting formats:



PeopleSoft Global Search Features

Search pages enable you to find and select data. Two types of searches can be performed on PeopleSoft search pages: Basic Searches and Advanced Searches.

Types of Searches

A **Basic Search** enables you to search by just one field at a time, and then only using the "begins with" operator. For a basic search, select the desired field from the Search By drop-down list box and then click the Search button to display the results of your search. You can narrow your search by first entering part of a value or description in the "begins with" text box. For example, if you want to find all queries beginning with the letters '*UHIR*', you would enter '*UHIR*' in the "begins with" text box and then click Search or press the ENTER key to process the search.

With an **Advanced Search**, you can further narrow your search, both by searching on multiple fields simultaneously and by using a variety of search operators.

The following search-related buttons appear on search pages of all types:

Search *Click to process the search.*

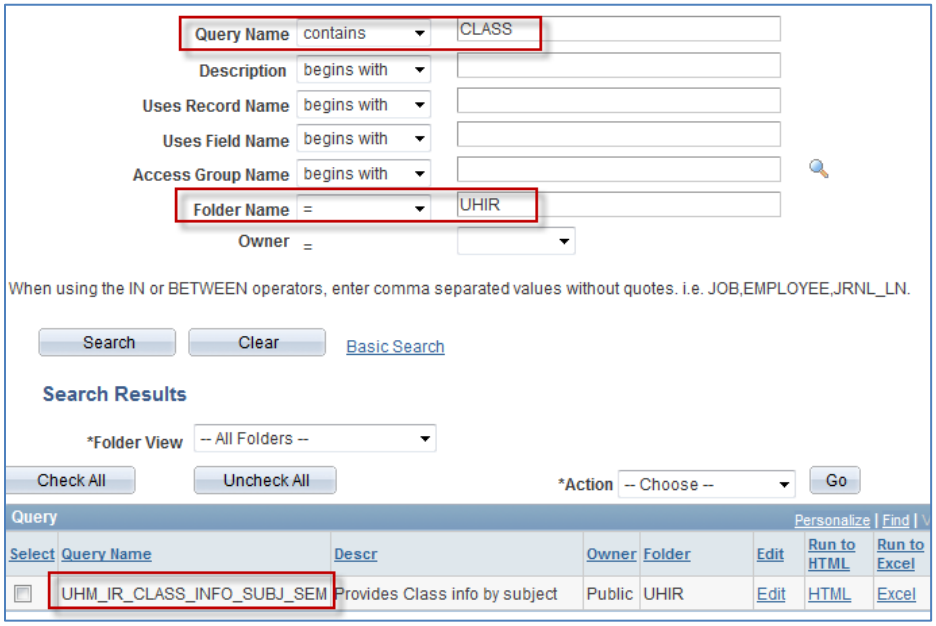
Clear *This button appears only for Advanced Searches. Click to clear entered text from all fields on a search page*

Search criteria can search for a full or partial value. The system queries the search record and presents a list of possible matches or, if only one match exists, displays the page that you requested.

If not all results appear at one time, you can click the Show Next Rows button (, the right arrow) in the grid header to view the next set of rows, and you can click the Show Previous Rows button (, the left arrow) to see previous sets of rows. You can also click the First and Last links to display the first and last sets of rows of search results. In addition, you might be able to click a View All or View *n* button to view all records at one time or to view a designated number of records.

Using Operators

When performing an Advanced Search, you can use a variety of Operators to narrow your searches. You can use the following operators: begins with; ***contains; =; not=; <; <=; >; >=; between; in*** (use a comma to separate the list of values, for instance *in 1000,1001*; to search for a character string that contains a space include that string within double quotes, for instance *in 1000," 1001"*). You can use an operator for more than one field to make your search even more specific. For example, you could narrow your search for queries where **Folder Name** = *UHIR* and where the **Query Name** contains *CLASS*.



Example Variations Using Search Operators:

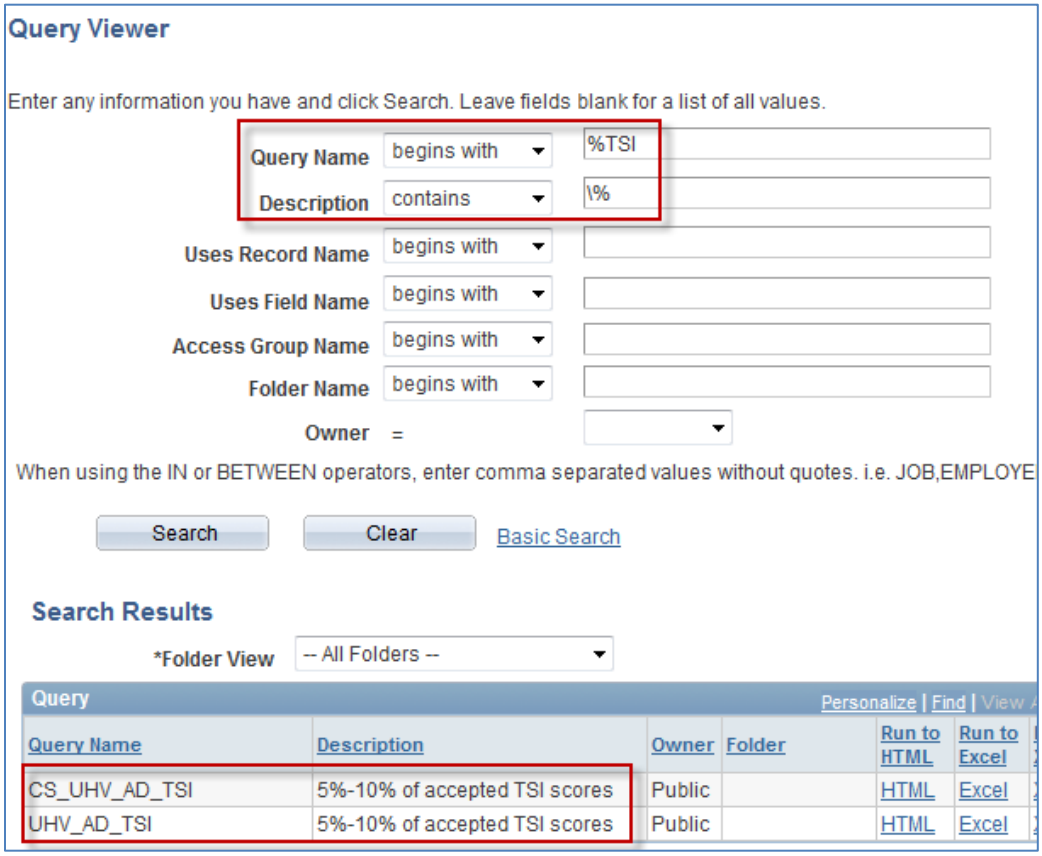
Search Operator	Operator Description	Sample Input Parameter	Sample Output Values Retrieved
begins with	Begins with value	"Smith"	"Smith, Smithee, Smithwick"
contains	Contains value	"Smith"	"Goldsmith, Smith, Smithee, Smithwick"
=	Equals value	"Smith"	"Smith"
not =	is Not Equal to value	"Smith"	All possible values excluding "Smith"
<	is Less Than; Comes Before	"Smith"	"Albert, Jones, Goldsmith, Nordstrom, Rainier"
<=	is Less Than or Equal to	"Smith"	"Albert, Jones, Goldsmith, Nordstrom, Rainier, Smith"
>	is Greater Than, Comes After	"Smith"	"Smithee, Smithwick, Turner, Urtz, Young"
>=	is Greater Than or Equal to	"Smith"	"Smith, Smithee, Smithwick, Turner, Urtz, Young"
between	is Between values	"Rainier" and "Smithwick"	"Rainier, Smith, Smithee, Smithwick"
in	is Included in the following list	"Jones, Smith, White"	"Jones, Smith, White"

Using Wildcard Characters

Wildcard characters can be used to help find data. PeopleSoft applications support three wildcard characters to help search for data in character fields. **Note**: wildcard characters only work with the "begins with" and "contains" operators.

PeopleSoft Query recognizes the **% (*percent*)** and **_ (*underscore*)** wildcards.

- **% (*percent*)** matches any string of zero or more characters. For example, '*C%*' matches any string starting with C, including C alone.
- **_ (*underscore*)** matches any single character. For example, '*_ones*' matches any five-character string ending with '*ones*', such as Jones or Cones.
- To use one of the wildcard characters as a literal character (for example, to include a % in your string), precede the character with a **\ (*backward slash*)** (for example, '*\%*'). The backward slash tells PeopleSoft to not treat the next character as a wildcard.

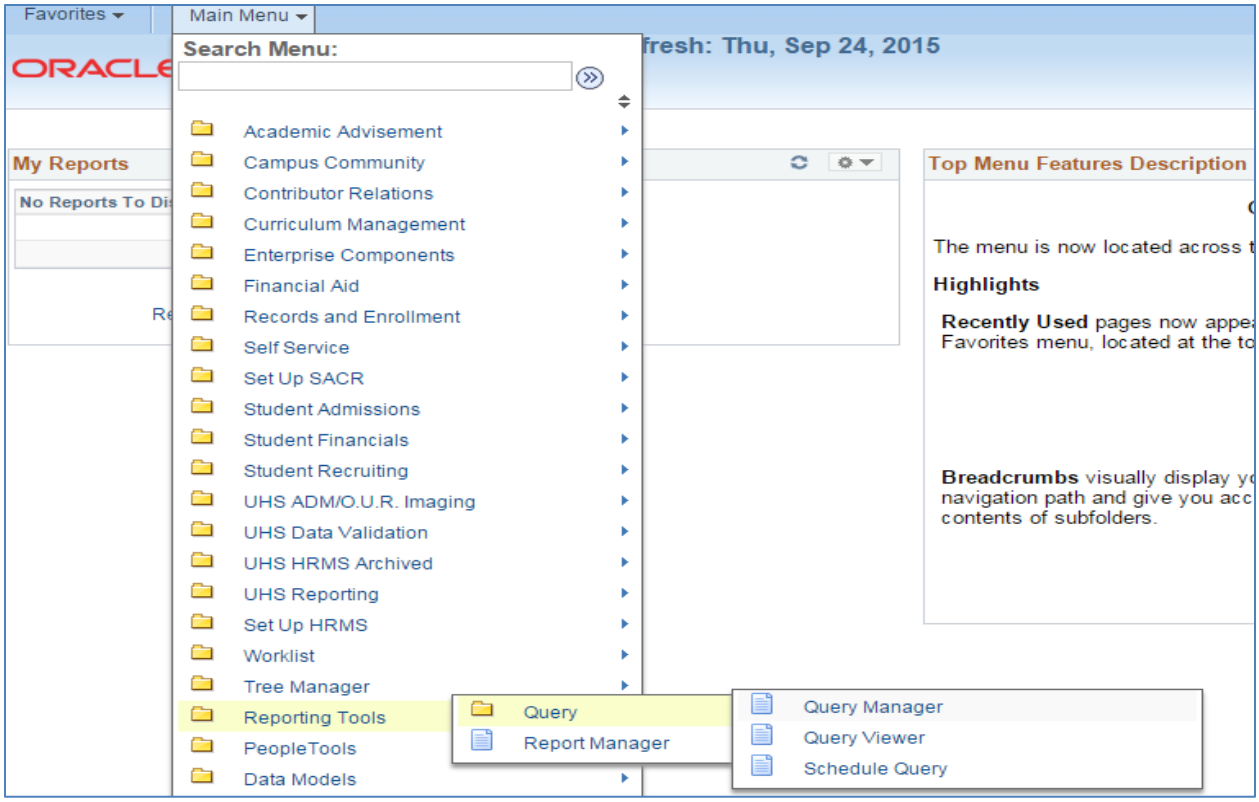


Searching for Queries

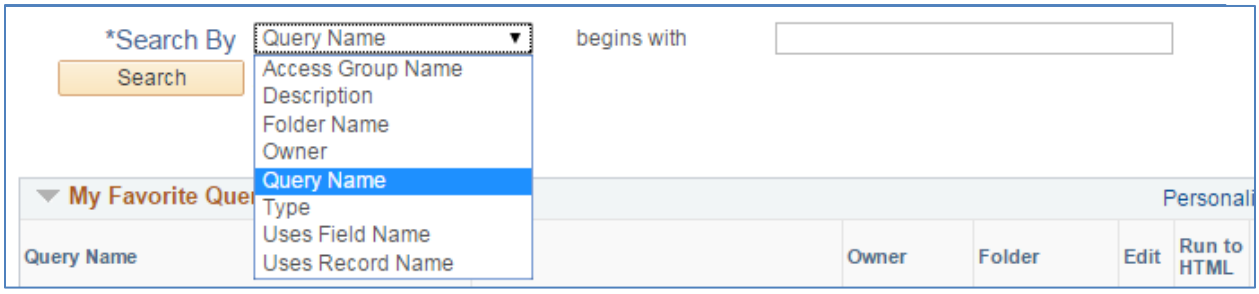
Both the Query Manager page and Query Viewer page enable a user to search for and run existing queries. *In addition to searching and running queries, Query Manager also allows the ability to create and edit queries.*

There are multiple options available for users to search queries; the most common are by query name, description, folder name, and owner. In addition, you can find a query if you know a record or field used by the existing query.

To get to Query Manager, click Main Menu > Reporting Tools > Query > Query Manager; to get to Query Viewer, click Main Menu > Reporting Tools > Query > Query Viewer.

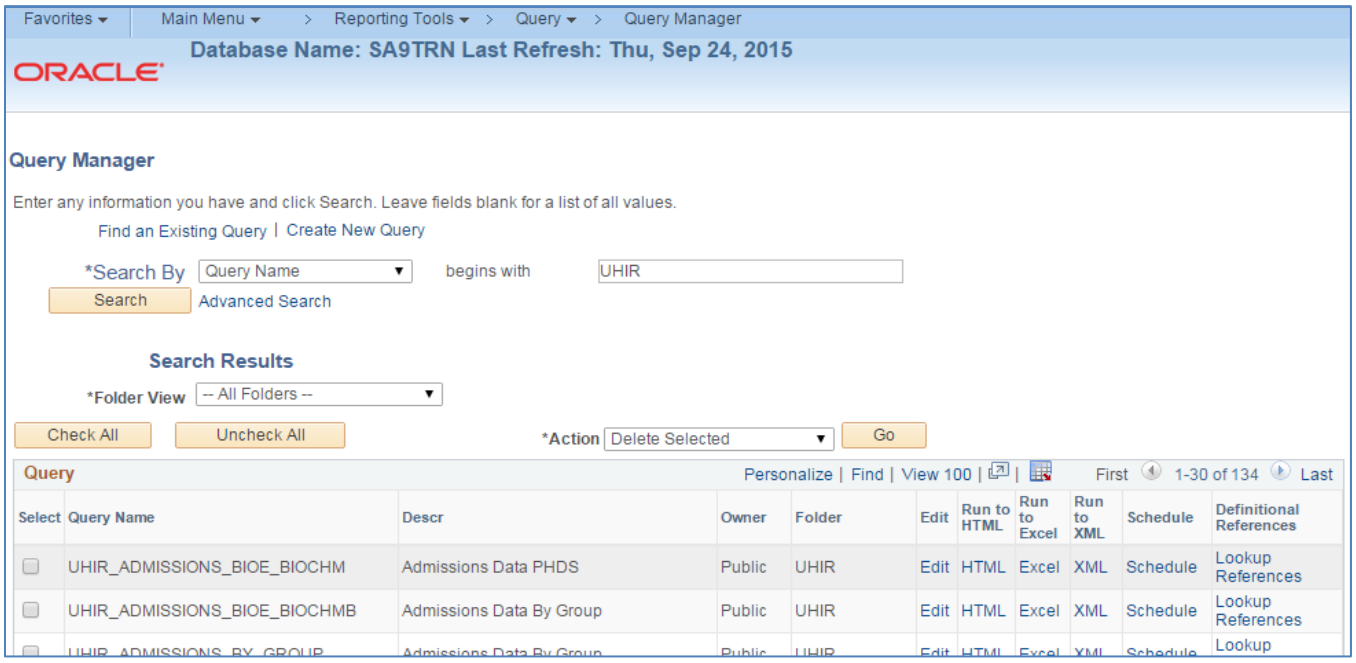


Queries can be searched using the following Search By parameters: Access Group Name, Description, Folder Name, Owner, Query Name, Type, Uses Field Name, and Uses Record Name.

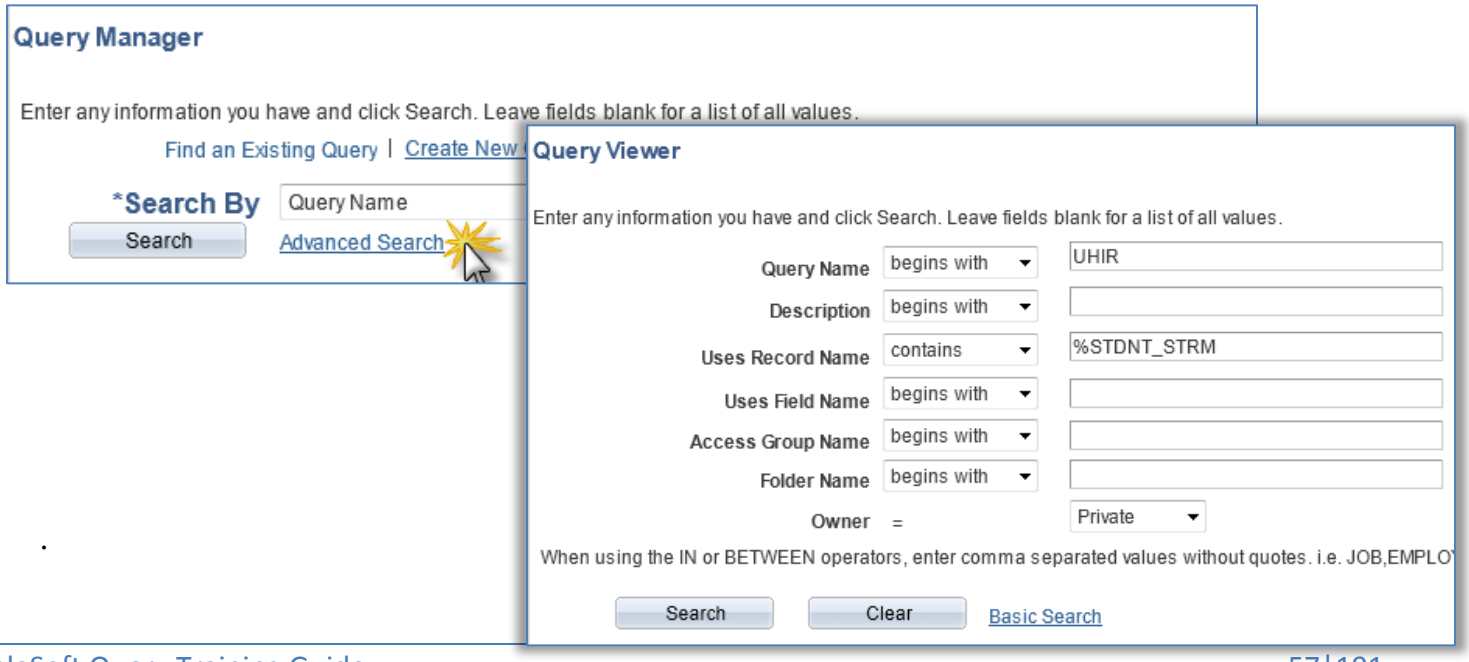


Both basic searches and advanced searches can be performed. The Basic Search allows you to search using the "begins with" condition. The Advanced Search allows you to perform a progressively narrower search by using multiple search criteria and selecting from a broad choice of conditions.

To perform a Basic Search, simply select an option from the Search By drop down list and then type in the text string to be searched. If you do not know the entire string (such as whole query name) simply enter the beginning characters of the text string as a partial string search; wildcards can be used in generating the search.

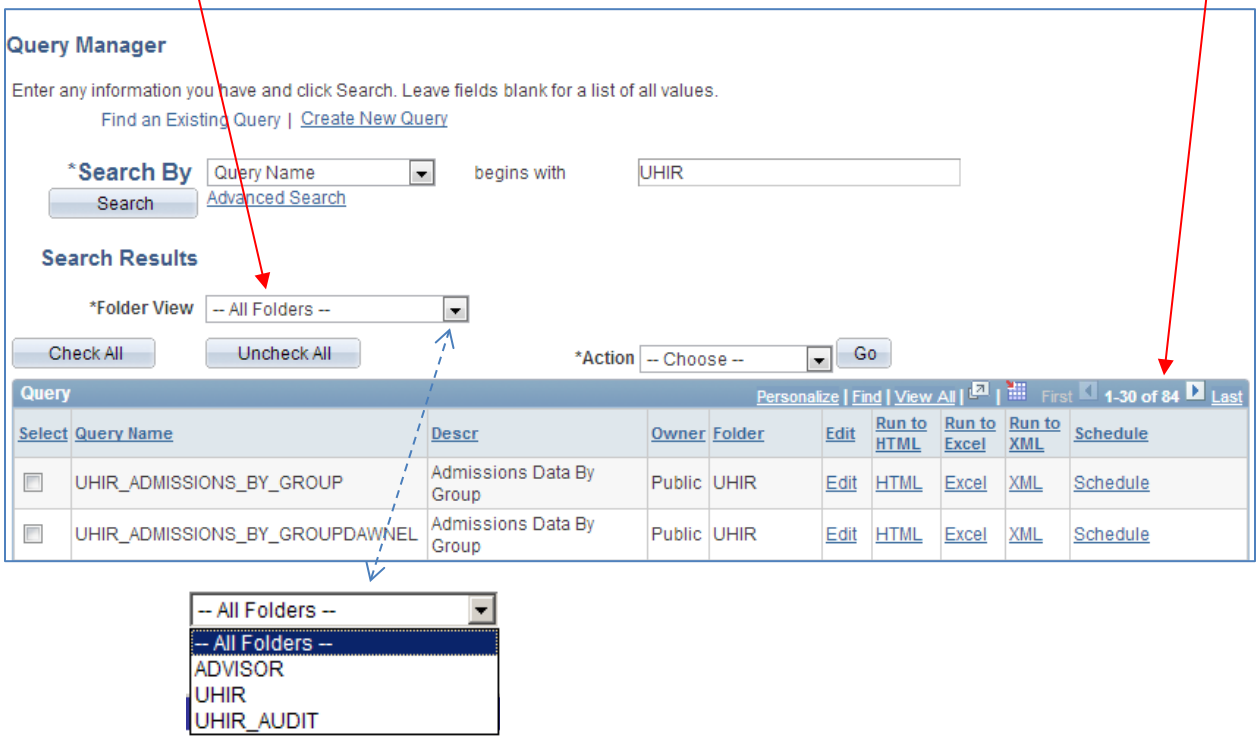


To perform an Advanced Search, click the Advanced Search link. On the Advanced Search page, select the appropriate Search By criteria and conditions, and then enter a search string in each of the corresponding fields. Click the Search button to display a list of queries that match your search criteria. The Search Results page appears. The results list all the queries that match the search criteria.



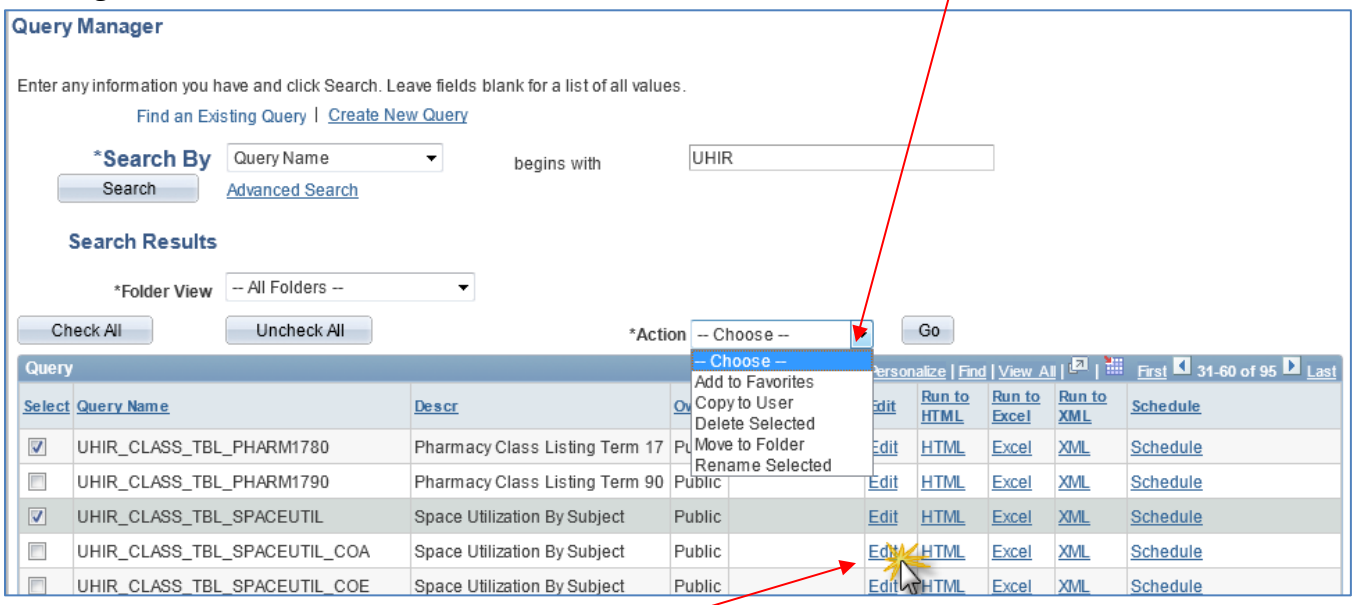
.

The Folder View drop down list enables the user to further refine the queries that are retrieved from the search criteria. The available folders in the drop down are based on the existing queries in Query Grid.



'Query Manager' Search Page Actions

PeopleSoft enables users to perform several query related actions by simply clicking on the Select checkbox next to one-to-many queries and selecting an option from the Action drop down list and clicking Go.



Users can modify the query by clicking the Edit link.

The following Actions can be performed on the queries selected:

Action	*How Used*
Add to Favorites	Adds the selected query to My Favorite Queries for easy access
Copy to User	Copies a private query to another user. You can copy only nonpublic queries to another user's list of queries. If the target user does not have permission to access all of the records in a copied query, that query does not appear in the target user's list of queries. When permission has been granted, the query appears in the list.
Delete Selected	Deletes the query. Users should ensure extreme caution in deleting queries and should generally ONLY delete PRIVATE queries.
Move to Folder	Organizing queries in folders can help you more easily access the queries. This action moves selected queries to a folder.
Rename Selected	Allows the user to rename the query. Users should ensure extreme caution in renaming queries and should generally ONLY rename PRIVATE queries.

Using Effective Data in Queries

Frequently there is a need to keep track of the history of changes to something in a database. The status of a student will change as he applies to a program, matriculates, possibly adds minors, and completes that program. There may also be a need to retain information about something, such as its name, flags, and amounts. In addition, to aid in planning ahead, there may be a need to store something that will come into effect in the future.

PeopleSoft uses special fields in many records to enable having data effective only at certain times – Effective Date (EFFDT), Effective Sequence (EFFSEQ), and Status as of Effective Date (EFF_STATUS). Whenever users add a row of data to the table, they specify the date on which that data becomes effective.

The most common use of effective date logic in queries is to report only active data current at the time the query is run. This may be referred to as the "maximum non-future effective date." Writing effective date logic manually involves adding a subquery on the same record as in the main query, joining on all key fields except for the date. This can be cumbersome for records with many fields in the key. Query Manager makes this unnecessary! When a record that has an Effective Date field is added to a query, an effective date criterion is automatically added to the query. You can change the criterion to look at the date in a field, expression, or a specific date of your choice. You can also show rows having the first effective date or last effective date regardless of whether the effective date is in the future. If the record also has an Effective Sequence field, the criterion will be "Eff Date <= Current Date (EffSeq = Last)," indicating that only the last row created on the effective date will be included in the results. You can edit this criterion to use the first row instead of the last or to show all rows regardless of Effective Sequence. If you accidentally delete a criterion on Effective Date, you can add one manually. If you choose EFFDT as a field in the criterion, you can choose from special Condition Types that apply only to effective dates:

- Eff Date < – effective date is less than the selected date
- Eff Date <= – effective date is less than or equal to the selected date

- Eff Date > – effective date is greater than the selected date
- Eff Date >= – effective date is greater than or equal to the selected date
- First Eff Date – effective date is the earliest for the key
- Last Eff Date – effective date is the latest for the key

The selected date can be the current date, a constant, a value in a field, or the result of an expression. Effective dates are tied to keys, so "First Eff Date" refers to the row having the earliest effective date for all rows having the same values in their key fields (except Effective Date and Effective Sequence). Query Manager does not automatically add criteria on Status of Effective Date. If you are interested in only active or only inactive rows, you must manually add the criterion on EFF_STATUS to your query. There are just two possible values of EFF_STATUS: 'A' for active and 'I' for inactive.

Running Existing Queries

Queries can be run from both Query Viewer and Query Manager. Running a query in these components will display the results in a new browser window. To run a query from either Query Viewer or Query Manager, click the HTML or Excel link to the right of the query on the Search Results page. From within Query Manager, you can also run a query by clicking on the Run page tab or Rerun Query link within the Run page.

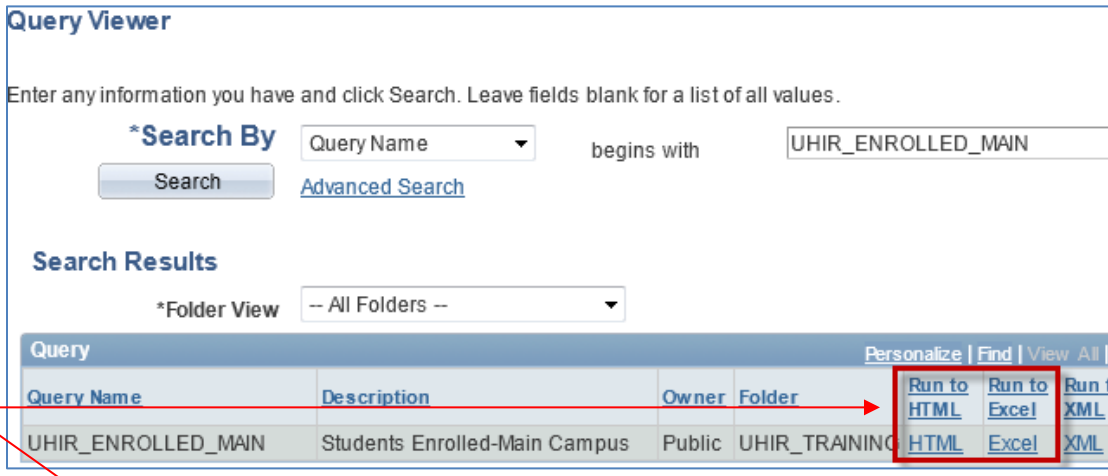
Practice Example: Find a list of students that are currently enrolled ("UHIR_ENROLLED_MAIN") at UH – Main and run the query.

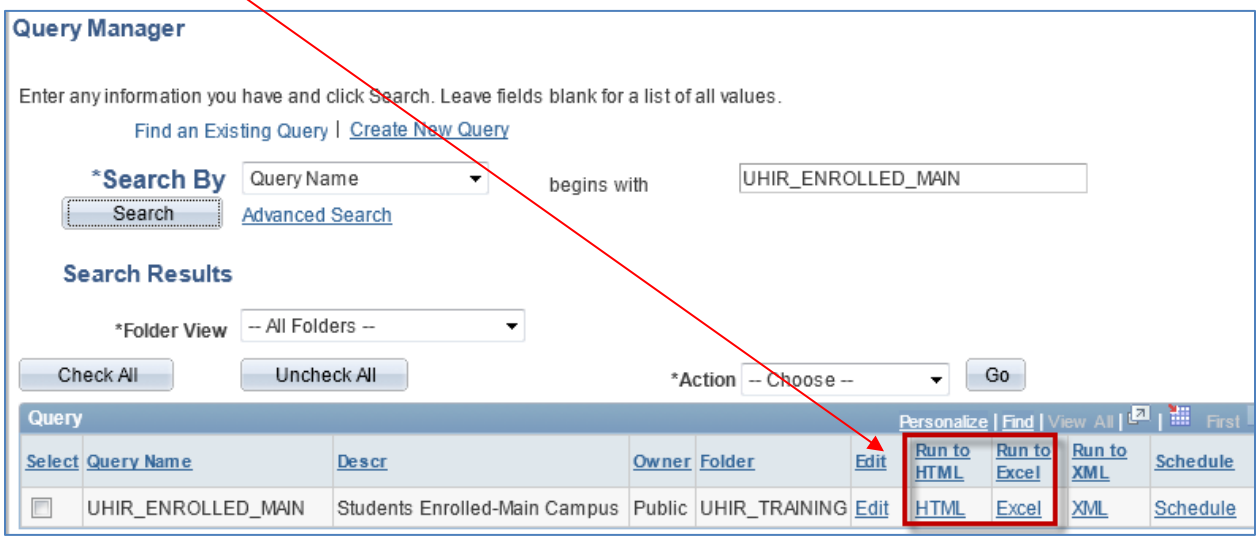
Steps:

(a) Navigate to *Reporting Tool, Query, Query Manager (Query Viewer)*.

(b) Search for query *UHIR_ENROLLED_MAIN*, and click Search.

(c) Click the HTML or Excel link





The expected results will be displayed in a new browser window.

(HTML)

View All First 1-40 of 40 Last

	ID	Name	FERPA	Email	EnrolledCurrent
1	1115268	Anderson,Joselyn Nashell	N	jnanderson4@uh.edu	Y
2	0491036	Anderson,Bessandra Traweek	N	btanderson@uh.edu	Y
3	0601649	Anderson,Christen Latrice	N	clanderson4@uh.edu	Y
4	0884355	Anderson,Kathryn L	N	klanderson@uh.edu	Y
5	0927009	Anderson,Claire E	N	ceanderson2@uh.edu	Y
6	0900270	Anderson,Holly A	N	haanderson@uh.edu	Y
7	0999255	Anderson,Kevin	N	kanderson6@uh.edu	Y
8	1067370	Anderson,Laurie N	N	lnanderson@uh.edu	Y
9	0983854	Anderson,Cassandra Brooke	N	cbanderson3@uh.edu	Y
10	1081571	Anderson,Jordan M	N	jmanderson8@uh.edu	Y

(Excel)

	A	B	C	D	E
1	Students Enrol	40			
2	ID	Name	FERPA	Email	EnrolledCurrent
3	1115268	Anderson,Joselyn Nashell	N	jnanderson4@uh.edu	Y
4	0491036	Anderson,Bessandra Traweek	N	btanderson@uh.edu	Y
5	0601649	Anderson,Christen Latrice	N	clanderson4@uh.edu	Y
6	0884355	Anderson,Kathryn L	N	klanderson@uh.edu	Y
7	0927009	Anderson,Claire E	N	ceanderson2@uh.edu	Y
8	0900270	Anderson,Holly A	N	haanderson@uh.edu	Y
9	0999255	Anderson,Kevin	N	kanderson6@uh.edu	Y
10	1067370	Anderson,Laurie N	N	lnanderson@uh.edu	Y

Running an Existing Query with a Prompt

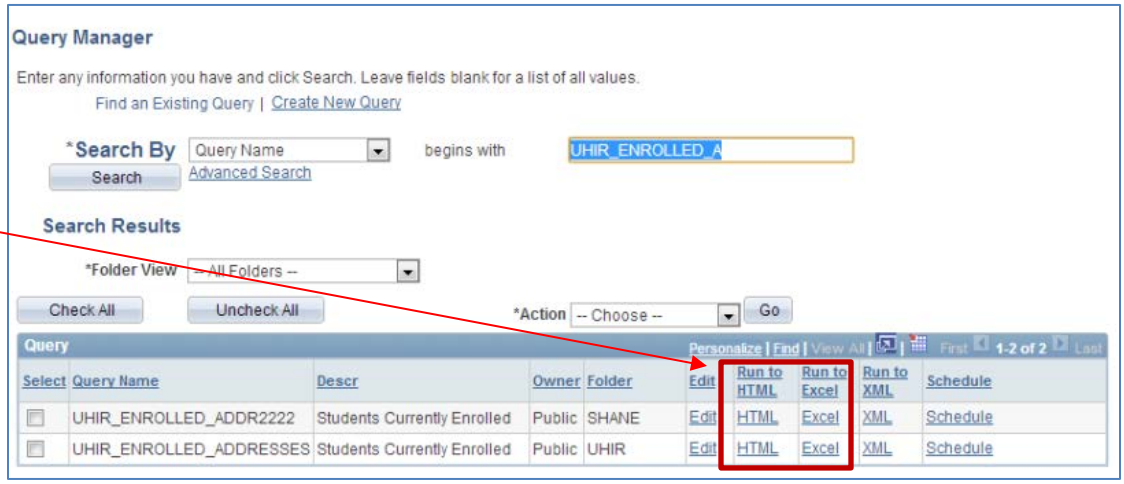
Some queries use run-time prompt variables when executed. Prior to retrieving the query results the query requires the user to enter values for certain criteria. You must provide values for the criteria.

Practice Example: Find list of addresses for students that are currently enrolled at UH – Main and run the query.

Steps:

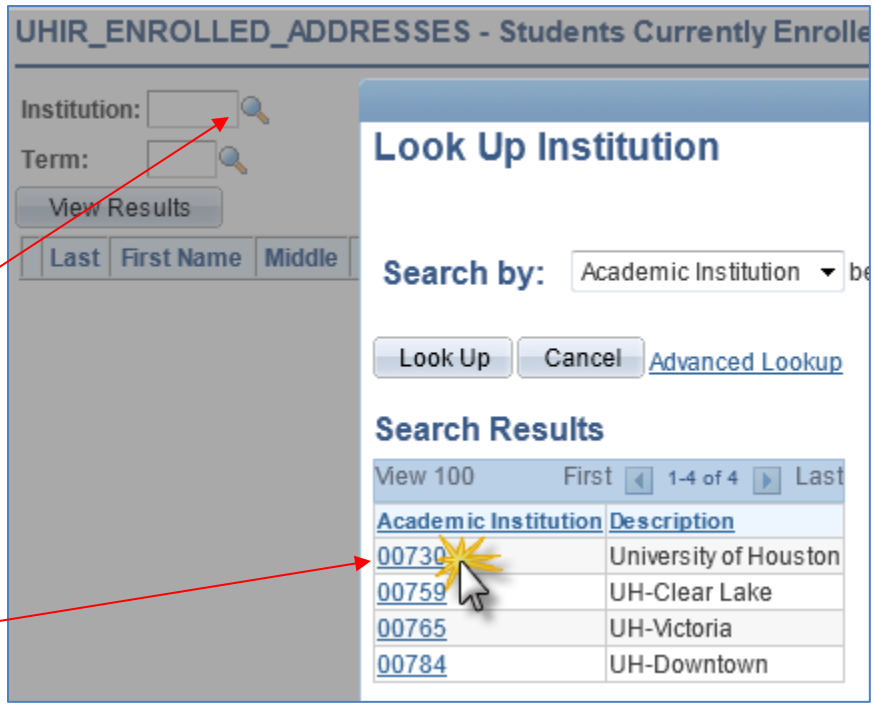
(a) Search for query *UHIR_ENROLLED_A*, and click Search.

(b) Click the HTML or Excel link



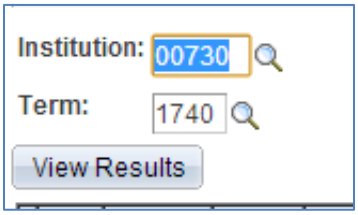
Steps:

Prompt box(es) will appear for fields needing run-time variables. Some prompt criteria may optionally provide a Lookup sub-search for the prompt value.



(c) Click on the looking glass to search for *UH – Main (00730)* then click on the desired value to select it into the prompts (value '00730').

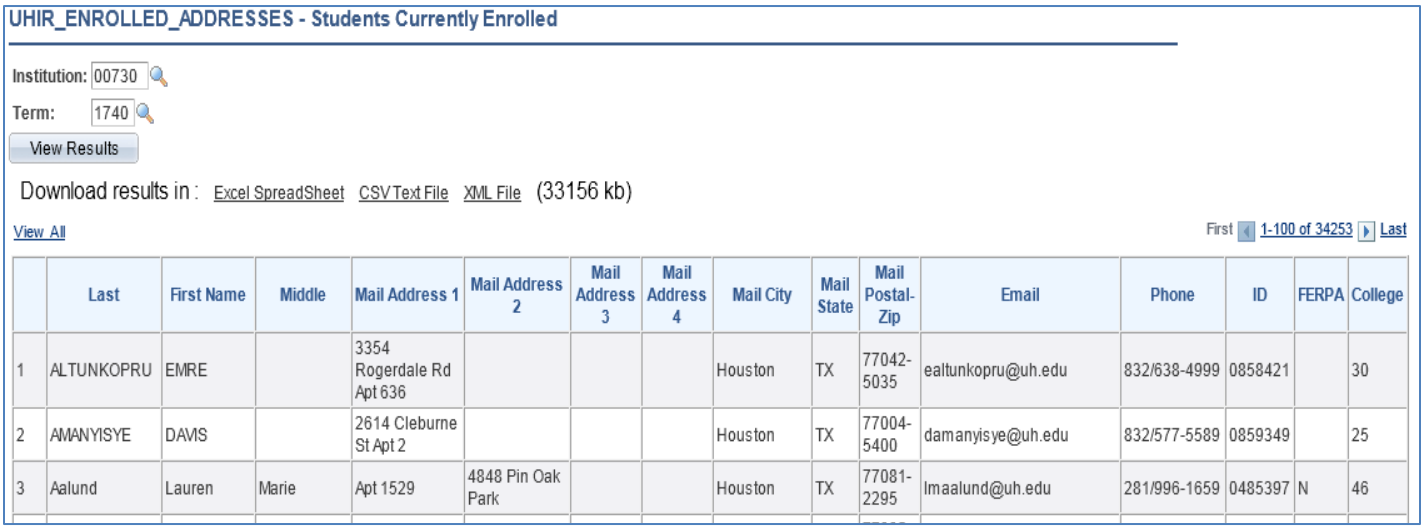
The institution code will appear in the prompt box.



(d) Type in term '1740' in the 'Term:' prompt box which is underneath Institution.

(e) Click on View Results to view addresses of UH-Main currently enrolled students.

.



	Last	First Name	Middle	Mail Address 1	Mail Address 2	Mail Address 3	Mail Address 4	Mail City	Mail State	Mail Postal-Zip	Email	Phone	ID	FERPA	College
1	ALTUNKOPRU	EMRE		3354 Rogerdale Rd Apt 636				Houston	TX	77042-5035	ealtunkopru@uh.edu	832/638-4999	0858421		30
2	AMANYISYE	DAVIS		2614 Cleburne St Apt 2				Houston	TX	77004-5400	damanyisye@uh.edu	832/577-5589	0859349		25
3	Aalund	Lauren	Marie	Apt 1529	4848 Pin Oak Park			Houston	TX	77081-2295	lmaalund@uh.edu	281/996-1659	0485397	N	46

Downloading Queries

You can download your query to a Microsoft Excel spreadsheet or CSV text file. If you download your query from the Run page, the query has a different default filename than if you download your query after clicking the HTML or Excel links. These default filenames are different because: (a) using the Run page to run queries, queries are run using the application server, and (b) using the HTML or Excel links, queries are run using a query service.

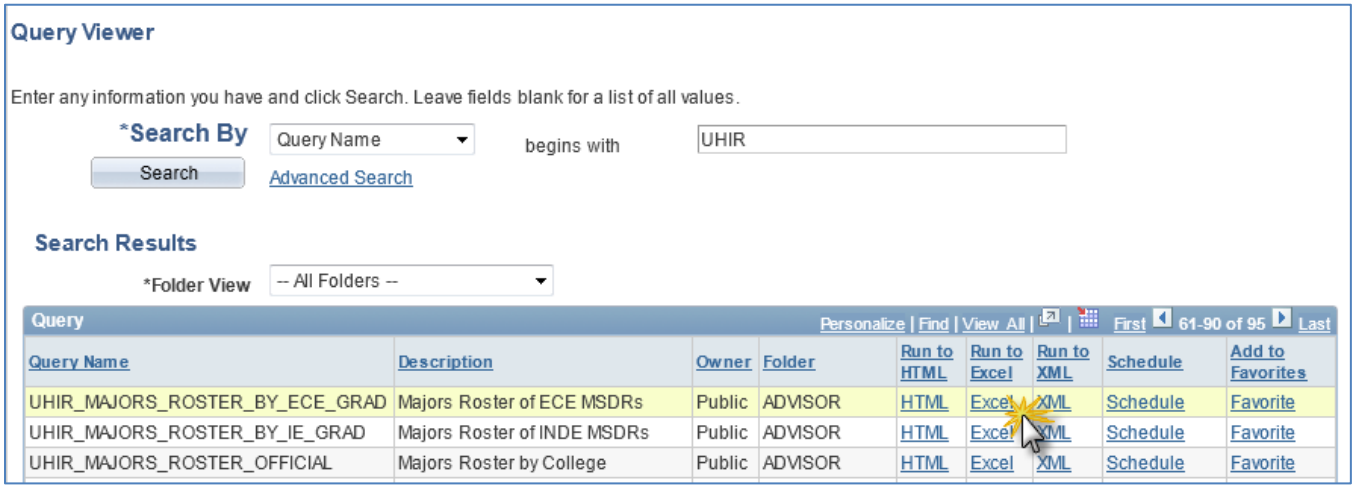
To download your query results, select one of the following:

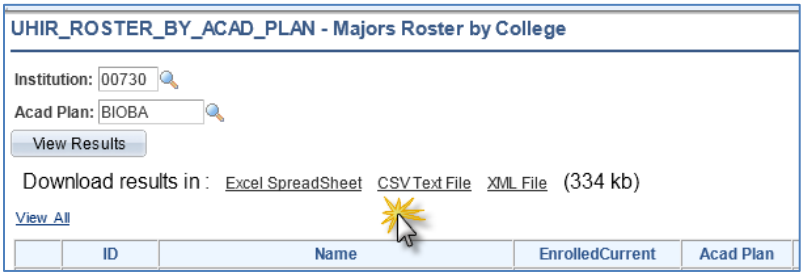
• Excel: Click this link on the Query Manager or Query Viewer search results page.

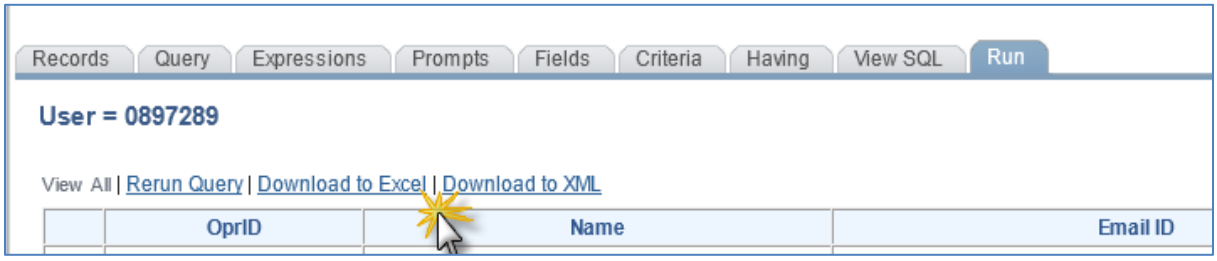
• Download to Excel: Click this link on the Query Manager or Query Viewer Run page.

• Excel SpreadSheet: This option is available after you have clicked the HTML link on the Query Manager or Query Viewer search results page. However, you can also click the Download to Excel or Excel links without downloading the query to HTML. You can configure your environment to open the Microsoft Excel file in a separate window or save it as a file on your local hard drive by modifying the File Type Option settings for Microsoft Excel Worksheets.

• CSV Text File: This option is available after you have clicked the HTML link. If you click this option, the File Download page appears, at which point you can open the file in your browser or save it to disk. The rows of output is not limited to the approx. 65k Excel 2007 limit.







Scheduling Queries

Query Viewer and Query Manager interact with PeopleSoft Process Scheduler to enable you to schedule queries to run at a later time; or to schedule queries with large results to run over the server instead of the web client that has limits on the retrieval of large results. You can schedule queries to run at predefined times or on recurring schedules (allowing the user to run the same query at regular intervals). The results of scheduled queries are retrieved in PeopleSoft Report Manager and/or Process Monitor.

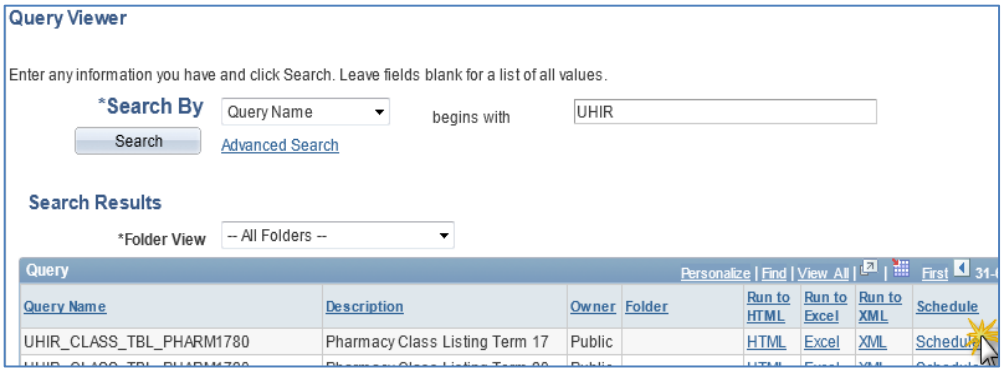
You can initiate scheduling of a query by clicking on the Schedule link from the Search Page of Query Viewer and Query Manager which takes you to the Schedule Query page; you can also initiate scheduling through navigation Main Menu > Reporting Tools > Query > Schedule Query. You can edit existing parameters for scheduled queries using the same steps.

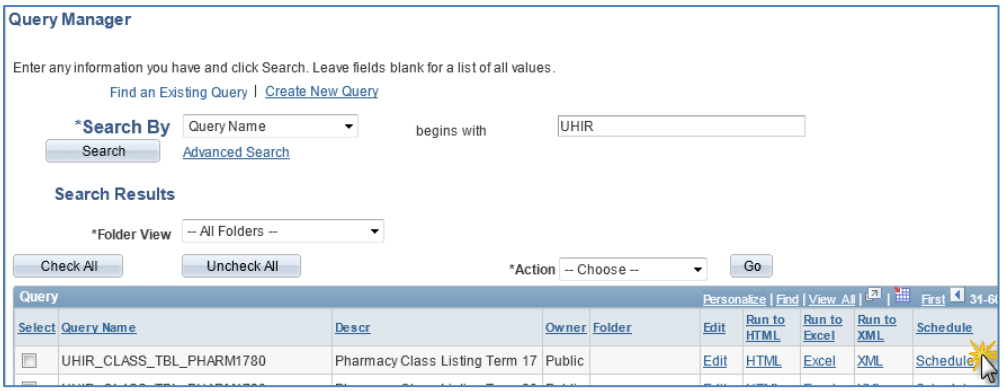
Queries are scheduled through submitting a Process Request. A process request enables you to submit a job or process to run. The Schedule Query page submits a process request and brings up the Process Request page, which enables you to specify such variables as where to run the process and in what format to generate the output based on a run control ID.

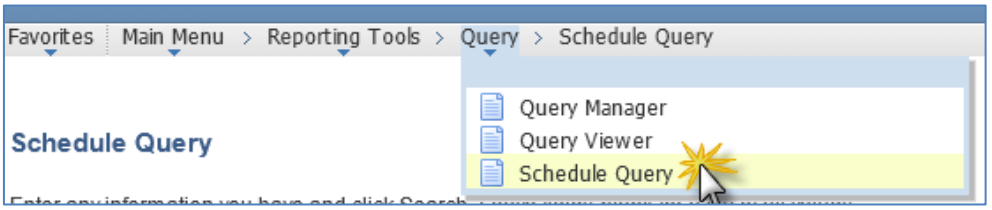
Practice Example: Schedule query UHIR_ENROLLED_MAIN to run every day at 10pm.

Steps:

(a) Initiate a query to schedule by either clicking on the Schedule link in Query Viewer or Query Manager; or by following navigation *Reporting Tool, Query, Schedule Query*.

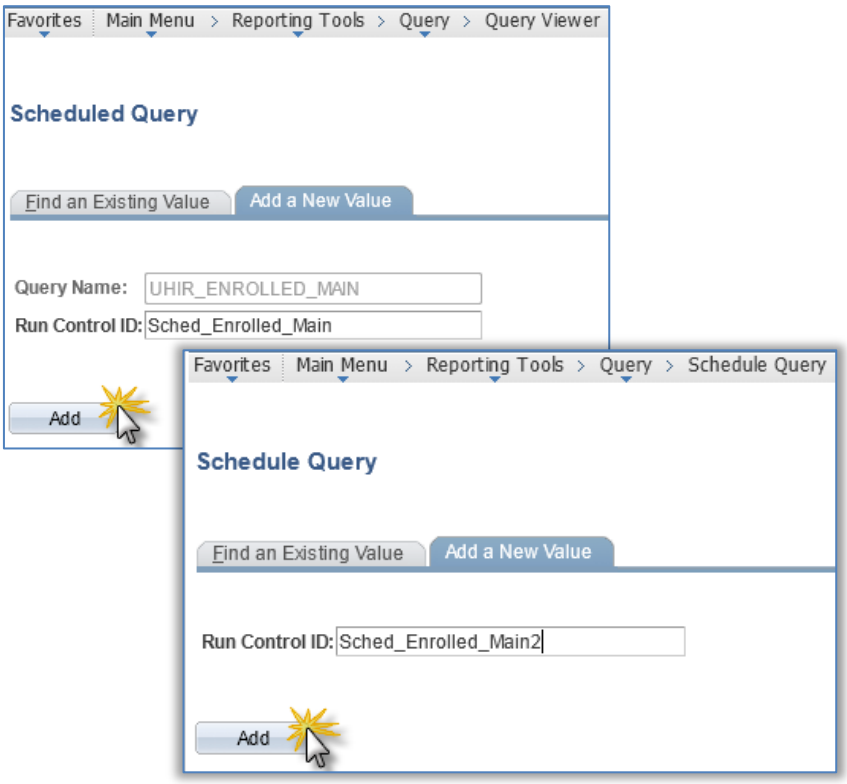






Steps:

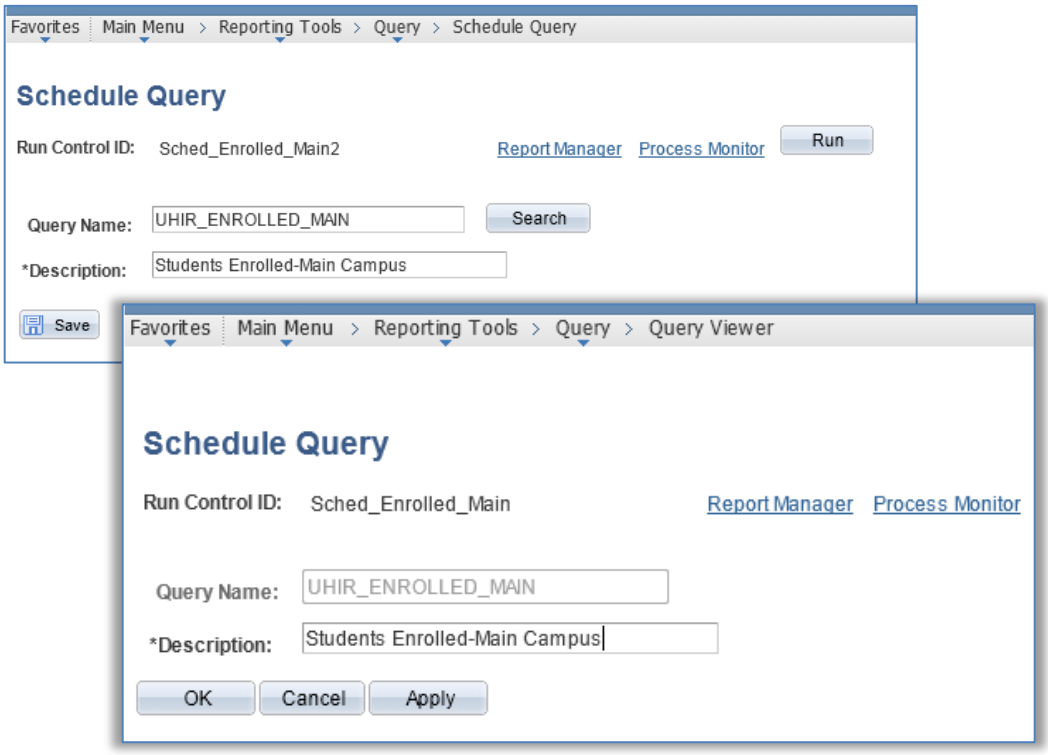
(b) Search for and select an existing run control ID, or select the Add New Value tab to enter a new ID. Click on Add to create the run control.



(c) If initiated from menu navigation Schedule Query, select the query for which you want to schedule; if initiated through Query Viewer/ Manager the query is already selected for you but you need to update the description field.

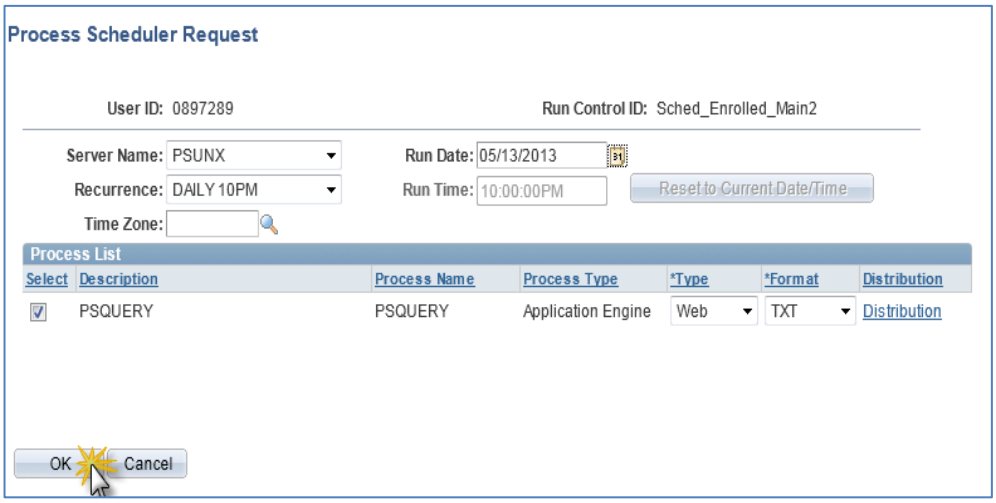
(d) Update any prompt parameters by clicking on the Update Parameters link. **Note**: if the query does not have any prompts, the Update Parameters link does not appear.

(e) To save changes and remain on the page, click the Save button if initiated through menu navigation Schedule Query, click the Apply button if initiated through Query Viewer/ Manager.

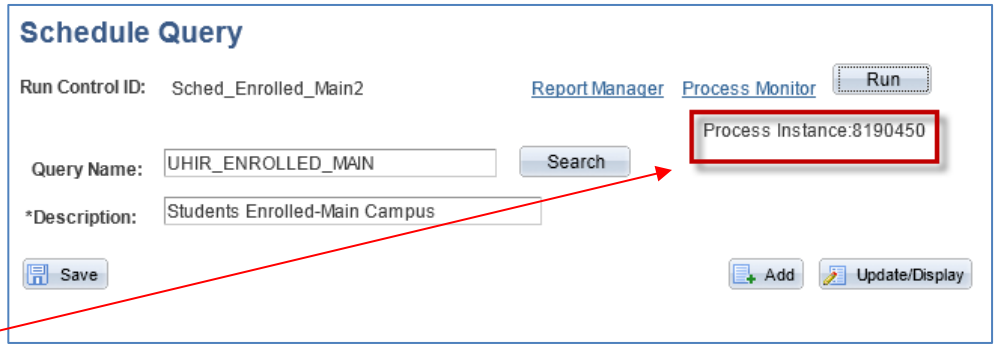


Steps:

(f) Click the Run button to submit the query process request to display the Process Scheduler Request page. When you are scheduling queries from Query Manager or Query Viewer, the Run button is replaced with an OK button.



(g) The Process Scheduler Request page enables you to set the server, run date and time, how often the process runs (the recurrence of the process), output type, and format. Complete the fields on the Process Scheduler Request page and click OK.



You will be returned to the Schedule Query page with a Process Instance number that has been assigned for the process request. To monitor the completion status of the request you can click on the Report Manager or Process Monitor link.

'Process Scheduler Request' Page Objects:

Page Object	*How Used*
Server Name	Select the name of the server on which you want the process to run. It's is suggested users select server 'PSUNX' for scheduling most queries; use 'PSNT' when saving it to a file directory (to the P: drive location).
Recurrence	If you wish to have the query run on a recurring basis, select the recurring time intervals for the process to run; i.e. "FRIDAY 6:15AM" meaning the query will run output every Friday at 6:15 in the morning.
Time Zone	Select the time zone in which the process will run.
Run Date	Select the date on which you want the process to run.
Run Time	Select the time at which you want the process to run.

Page Object	*How Used*
Reset to Current Date/Time	Click to reset the run date and time to the present date and time.
Type	Select the output type for this job or process: ***Web** (Commonly used)**:*** Sends all output of the process to the report repository, including log and trace files. The format of the report is specified by the format list. ***Window** (Commonly used)**:*** Sends the output to a new browser window. The status of the process now appears in the new browser window before the results are displayed. All output for the process is also sent to the report repository, including log and trace files. The format of the report is specified by the format list. ***Email** (Commonly used)**:*** Sends the output through an email. To distribute a report to an email list, enter the appropriate information on the Distribution Detail page by clicking the Distribution link. By default, the output is sent through email to the person running the process.**Note.** Using Scheduled Query, you can schedule queries to run and have the results email directly to recipients as an attachment. Email should **NOT BE USED** when the query output includes **PROTECTED DATA.** ***File:*** Writes the output to the file that you indicate in the Output Destination field. **Note.** If you set Type value to *File* or *Printer*, you must enter the name of the directory in the Output Destination field. ***Printer:*** Sends the output to a printer. You can enter a custom printer location in the Output Destination field if you have the appropriate security access. If the Output Destination field is left blank, the printer that is defined on the Process Profile Permissions page is used. If that printer is undefined, the default printer that is defined for the process scheduler is used
Format	Select the output format for this job or process. Suggested formats are 'TXT' for using both native text files as well as in Excel; 'XLS' for Excel use (limited to Excel 2007 maximum row/data limits); 'PDF'; and 'HTM' for html/window.
Distribution	Click the Distribution link to access the Distribution Detail page, where you enter additional distribution information when the output type is ***Web,* Window, or *Email.*** Email Only To email a link of the report to users, click the "Email Web Report" checkbox and add additional email addresses to the 'Email Address List' textbox; multiple addresses must be separated by a semicolon. You can add an email subject and text to add to the results. This will result in the email including a link to the query results in the 'Report Manager'. Recipients of the email can click on the link in the email which will take the

Page Object	*How Used*
	user to the process instance. Distribute To The default distribution is only to the user that runs the query. If additional users need to view the query in the 'Report Manager', add them to the "Distribute To" section. Any users selected here will be able to view the query in the Report Manager. Click the 'OK' button after setting distribution options.
Output Destination	Enter the file directory path or printer destination for the output. For example, *C:\Documents and Settings\admin\psft\pt\8.50\appserv\prcs\QEDMO\log_output.* **Note.** This field is available only when the output type that you select is *File* or *Printer.* If you select an output destination (OutDest) for a process at the process definition level, this field is populated with that output destination.

Report Manager

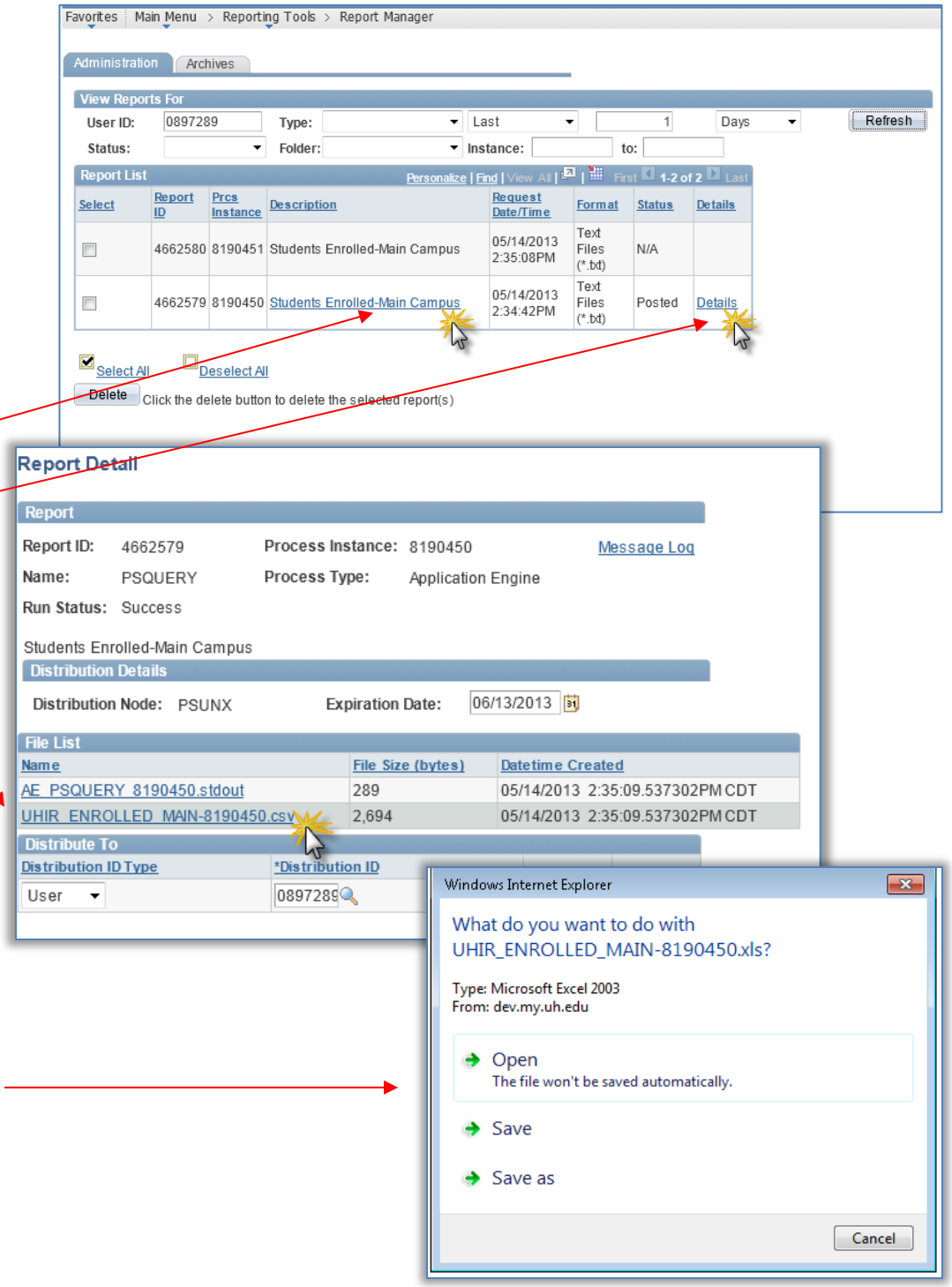
The results of scheduled queries can be retrieved in PeopleSoft Report Manager. Report Manager is like your own personal inbox of reports and process output.

Steps:

(a) To access the Report Manager navigate menu Reporting Tools > Report Manager, or click the Report Manager link on the Schedule Query page.

(b) To run the report from the Report Manager page, click on the report item's hyperlinked Description. To run the report from the Report Detail subpage click on the Details link and then the file output link (i.e. CSV file link).

(c) Open the file or save it to a desired location.

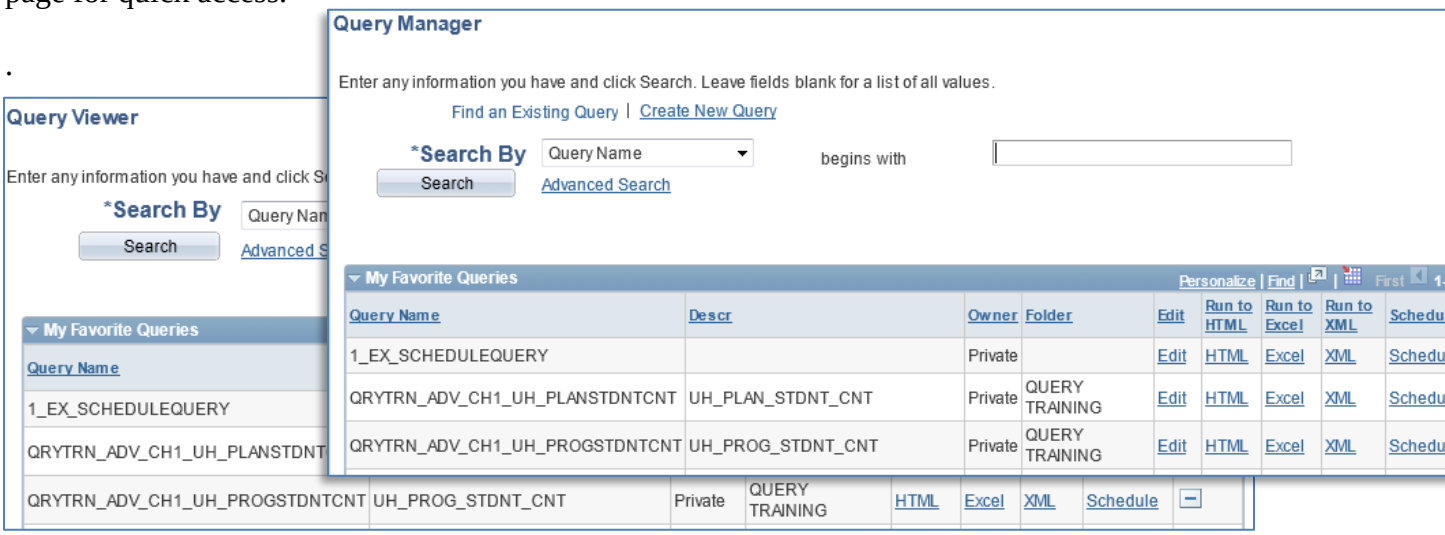


My Favorite Queries List

The Query Viewer and Query Manager Search pages offer a 'My Favorite Queries' section listing queries that have been tagged as favorites for quicker retrieval and execution of the queries. If you use certain queries often, you can put the queries in this list for easy access. You can add a query to the My Favorite Queries list from both Query Viewer and Query Manager.

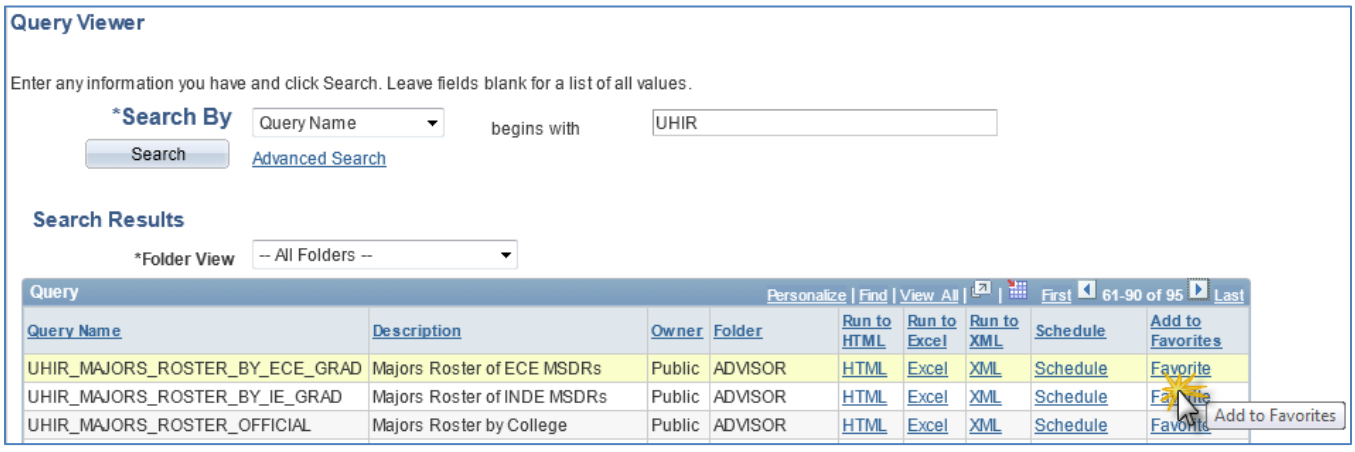
When you initially navigate to the query tool, the My Favorite Queries section appears at the top of the page for quick access.

.

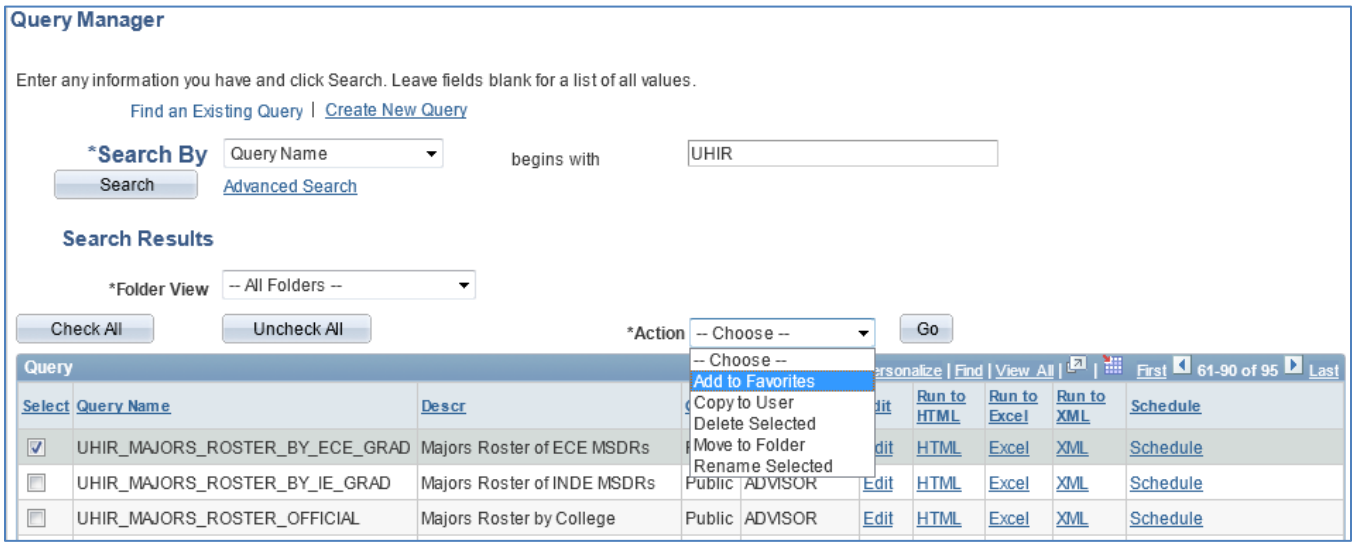


Steps:

(a) To add a query to the My Favorite Queries list from Query Viewer click on the Favorite link next to the desired query from the search results.



(b) To add a query to the My Favorite Queries list from Query Manager select the checkbox next to the desired query (you can select multiple query checkoxes), select Action 'Add to Favorites', and then click the Go button.



VOLUME II:

Basic PeopleSoft Query Writing

Objectives

Participants will learn the following concepts and procedures:

- Understand general Query Methodology
- Understand PeopleSoft Query Structural Process Flow
- Query Manager Design Pages and Page Links
- Understand how to develop simple queries
    - Selecting data from a single table
    - Specify the column order
    - Specify the sort order
    - Specify criteria for retrieving data

Overview

General Query Methodology and PeopleSoft Query Structural Process Flow will be discussed.

Pages and objects used for designing PeopleSoft queries will be explored. Users will then develop, save and run a basic query. Users will learn to develop simple queries by selecting from a single record, changing field headings, adding criteria, and sorting the records.

CHAPTER 4 – Query Writing Essentials

Overview In this chapter, you will learn essentials of Query Methodology and PeopleSoft Query Structural Process Flow.

Query Methodology

When creating queries, always endeavor to follow the below basic methodology:

1. Identify What Information is Really Needed
2. Determine Criteria Logic
3. Use Appropriate Records, Tables, and Fields, Column Ordering, Field Sorting
4. Perform Table Dumps to Learn Tables
    a. Identify Key Fields
    b. Develop Criteria for Table
    c. Identify Example/Sample Data
5. Create Table/Record Joins
    a. Identify needed join type
        i. Inner/Standard Join
        ii. Outer Join
    b. Run query after each new table join to compare what has changed – add/lost rows/data.
6. Verify Data Set
    a. Is this the data you want to use?
7. Determine Use of Advanced Functions
    a. Using Expressions
        i. Identify Data Type: Numbers, Characters, Date, Drilling URL
        ii. Identify Needed Manipulation: Data Type Conversion, Totals, Grouping, If–Then Logic
    b. Using Subqueries
        i. What table(s) and criteria in the subquery are needed?
        ii. What single field/expression of output needs to be returned?
    c. Using Unions
        i. What separate query results do you want to merge into a single result set?
        ii. Can the separate queries be merged to have the same number of fields, same data types, and same display order?
8. Build & Test in Increments
9. Using/Viewing SQL statement if troubleshooting is needed
    a. What's really going on in the background?

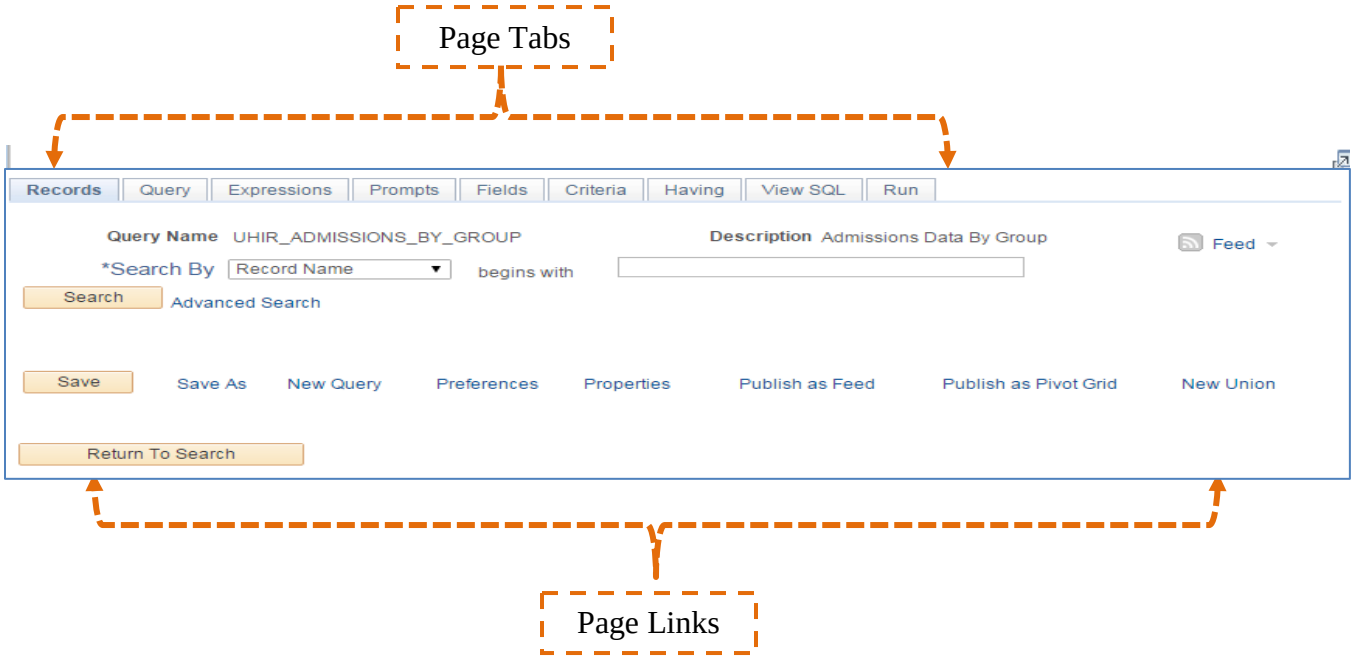
PeopleSoft Query Structural Process Flow

The below process flow illustrates the basic steps in constructing a PeopleSoft Query:

CHAPTER 5 – Query Manager Design Pages and Page Links

Overview In this chapter, you will learn to navigate the pages used in designing PeopleSoft queries.

Several pages are used in constructing PeopleSof t Queries. Some pages are accessed via tabs located at the top of Query Manager and some are accessed via links, icons or buttons nested within other pages.



Summary Page Descriptions:

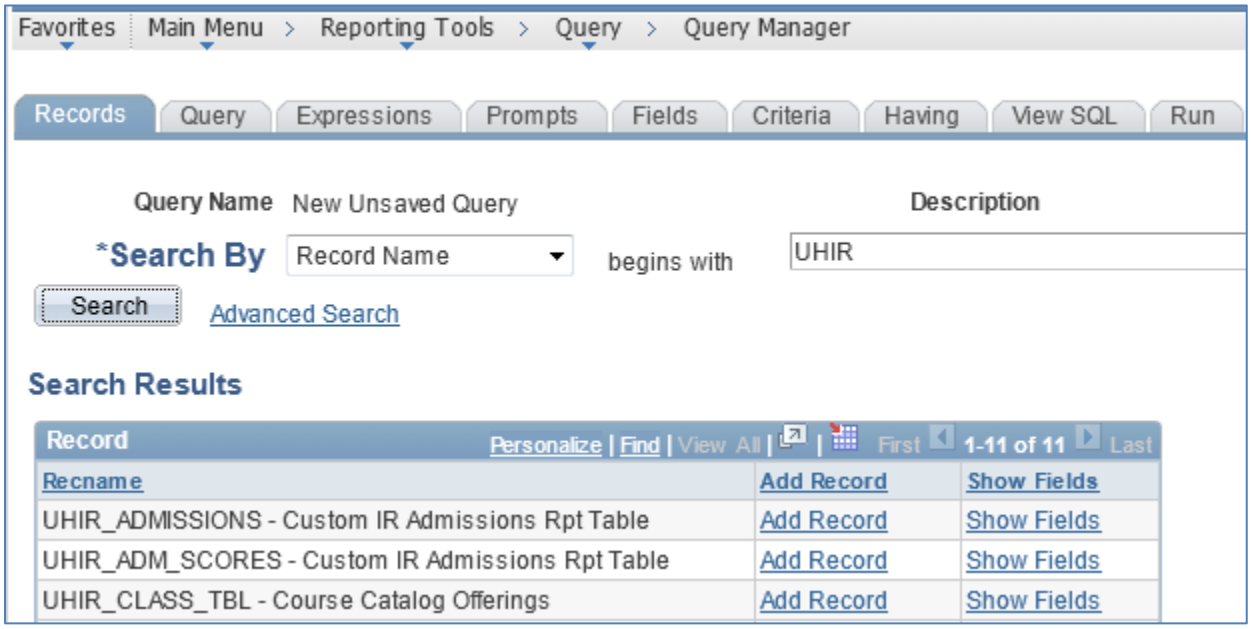
Page/Page Link	*How Used*
Records	Select the records upon which to base the query.
Query	Add fields to the query content and navigate to Criteria page. You can also add additional records by performing 'Hierarchical' and 'Related Records' joins. When you first access this page, if you have selected the record for an effective-dated table, a page informs you that an effective date criteria has been automatically added for this record. Click OK to close the page.
Expressions	Expressions are calculations that PeopleSoft Query performs as part of a query. Use them when you must calculate a value that PeopleSoft Query does not provide by default—for example, to add the values from two fields together or to multiply a field value by a constant. You can work with an expression as if it were a field in the query: select it for output, change its column heading, or choose it as an "order by" column. In Query Manager, you can use expressions in two ways: • As comparison values in selection criteria. • As columns in the query output.

Page/Page Link	*How Used*
Prompts	Add or edit variables used at run-time.
Fields	View how fields are selected for output; view the properties of each field; change headings and aggregate values; delete fields from output; navigate to Edit Field Ordering page, Edit Field Properties page, and Criteria page.
Edit Field Ordering	Use to change the column order and row sort order for multiple fields.
Edit Field Properties	Use to format the query output for example column headings, display translate table values in place of codes, and aggregation.
Criteria	View and edit selection criteria to filter the data for output.
Having	SQL does not support the use of aggregate functions in WHERE clauses. Therefore, after you have applied an aggregate function to a field, you cannot use that field in your selection criteria, which corresponds to a SQL WHERE clause. When you want to select rows based on the results of an aggregate function, Query Manager enables you to create HAVING criteria. You might use such criteria, for example, when you want a list of students who have a count of degrees greater than two. In SQL, a HAVING clause is similar to a WHERE clause for rows of data that have been aggregated into a single row of output. The system evaluates WHERE clauses by looking at the individual table rows before they are grouped by the aggregate function, and then it evaluates HAVING clauses after applying the function. When you click the Add Criteria icon from the Fields or Query pages for an aggregate field, new criteria is added to the Having page instead of the Criteria page. Add selection criteria using the Having page in the same way that you add selection criteria using the Criteria page. Keep in mind that PeopleSoft Query compares the result of applying the aggregate function to the comparison value.
View SQL	View the underlying SQL code that Query Manager generates based on how you have designed the query. Viewing the SQL statement can be useful in troubleshooting queries that utilize records that have underlying joins associated with Security Search records. You cannot modify SQL on this page.
Query Preferences	Specify query preferences such as auto joins and auto preview.
Query Properties	View and edit data about the current query such as the query name, description, and distinct parameters.
Save Query / Save Query As	Save the query.
New Query	Click to access the PeopleSoft Query Manager where you can start creating a new query.
New Union	Click to create a union of multiple queries. This link is available only when query does not have union.
Delete Union	Click to delete an existing union of multiple queries. This link is available

Page/Page Link	How Used
	only when query have a union attached.
Return to Search	Click to return to the Query Manager search page where you can create new queries, modify existing queries, schedule queries, and organize queries.

'Records' Page

In your PeopleSoft database, tables are represented as record definitions. In PeopleSoft Query, we refer to the record definitions as records. The Records page allows the user to select those records (aka tables) to be used in the query. Following the concepts from the chapter on query searches you similarly search for the records to add to your query design by performing a partial search, full search, and/or advanced search of the record name, record description or fields contained in the record. Click on the Add Record link next to records you want to add to the query design.



'Records' Page Objects:

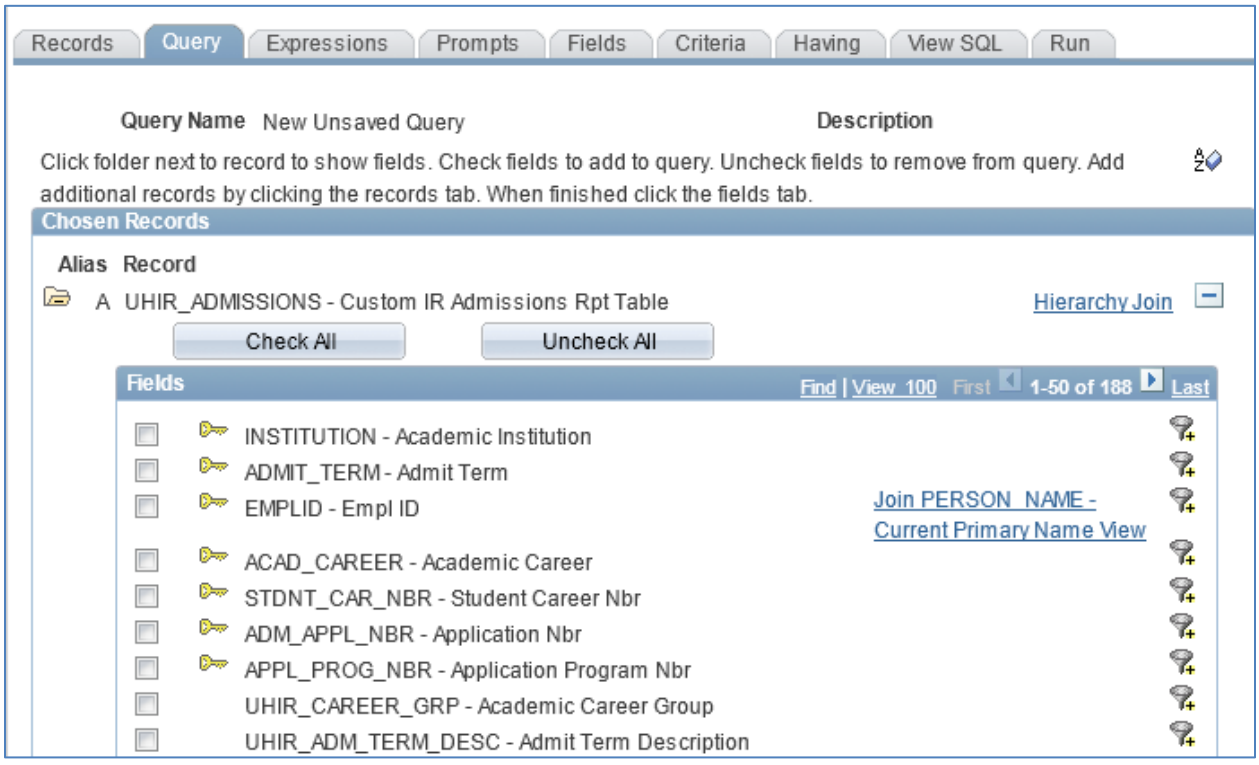
Page Object	How Used
Record	Identifies the record name and short description for the record.
Find	Click the Find link to search for a record within the list of retrieved records. The list will jump to the first record meeting the search string.
View 'nn'/ View All	By default, only a subset of items are included in the display range. To increase the range of items displayed (to show 50, 100, all), select the View 50/100/All link.
First	Click the First link to jump to the first subset of displayed items when scrolling through the display groups.
and	Show Next Rows/Show Previous Rows scroll bars. Use the left and right arrow scrollbars to go through the rest of the grouped items displayed in a

Page Object	How Used
	range.
Last	Click the Last link to jump to the last subset of displayed items when scrolling through the display groups.
Add Record	Click the Add Record link next to the record you want to add to the query design. This takes you to the 'Query' page, from which you can select which fields from the selected record to add to the query output.
Show Fields	Click the Show Fields link to preview the fields contained in the record. You can use this information to verify whether you want to include the record in the query design. Click 'Return' when you have finished previewing the fields.

.

'Query' Page

The Query page allows the user to select fields from the record that was selected on the Records page. Select the fields to add to the query either by clicking their check boxes or by clicking the Check All Fields button. When you have selected the desired fields, select the Fields tab.



'Query' Page Objects:

Page Object	How Used
	Click the Sort button once to list fields in alphabetical order. Click the button again to return to the original sort.
	Indicates Key Fields.

Page Object	*How Used*
	Click the Delete button to delete the row. A confirmation message appears. Click the Yes button to proceed with the deletion. Click the No button to cancel the deletion.
	Click the Use as Criteria /Add Criteria button to open the Edit Criteria Properties page, where you can determine how this field will be used as a criterion for the current query.
	From the Query tab, click the Folder button to view the fields for the chosen record, if they are not already displayed. Query Manager expands the record so that you can see the fields and make sure that this record has the content that you want. Click the Folder button again to hide the fields for a record. A key is displayed to the left of key fields.
Alias	The Alias name is the identifier that the system automatically assigns to the chosen records when building the SQL statement that is generated in the background.
Hierarchy Join	Click the Hierarchy Join link to join a child table to its parent table.
'Check All'	Click this button to check all fields in the record. After you select a field, the system automatically adds it to the query content and you can view it on the Fields page. This button does not appear when the field names are hidden.
'Uncheck All'	Click this button to clear all fields in the record.
Fields	Select the check box to the left of each field that you want to add to your query content.
Find	Click the Find link to search for a field within the selected record. The list will jump to the first field meeting the search string.
View 'nn'/ View All	By default, only a subset of items are included in the display range. To increase the range of items displayed (to show 50, 100, all), select the View 50/100/All link.
First	Click the First link to jump to the first subset of displayed items when scrolling through the display groups.
 **and** 	Show Next Rows/Show Previous Rows scroll bars. Use the left and right arrow scrollbars to go through the rest of the grouped items displayed in a range.
Last	Click the Last link to jump to the last subset of displayed items when scrolling through the display groups.
Related Record Join	Click these links to auto join two records that are based on a shared field. For example, in the above screen shot, the ACAD_CAR_TBL record is related to the UHIR_STDNT_STRM record by the ACAD_CAREER field.
Expand All	Click this button to view all fields in the records. This button appears only

Page Object	*How Used*
Records	when there is more than one record listed.
Collapse All Records	Click this button to hide all fields in the records. This button appears only when there is more than one record listed.

'Expressions' Page

Expressions are calculations that PeopleSoft Query performs as part of a query. Use them when you must calculate a value that PeopleSoft Query does not provide by default—for example, to add the values from two fields together or to multiply a field value by a constant. You can work with an expression as if it were a field in the query: select it for output, change its column heading, or choose it as an "order by" column. The result of an expression can be text, a date, or a number, and may be displayed and used in criteria as if it were a real field.

In Query Manager, you can use expressions in two ways:

- As comparison values in selection criteria.
- As columns in the query output.

Fields from records can be used in expressions by using their actual name, not their description and preceding the name with their alias and a period. Prompts can be used in expressions by using a colon followed by the prompt number, such as ":1".

One type of calculation that can be performed is mathematical. Standard mathematical order of operations (multiplication and division are done before addition and subtraction; computation goes from left to right) is used. Parentheses can be used to group calculations together and force them to be done out of this order. You can add, subtract, multiply and divide numeric values. The mathematical operators are:

- **+** (addition)
- **-** (subtraction)
- ***** (multiplication)
- **/** (division)

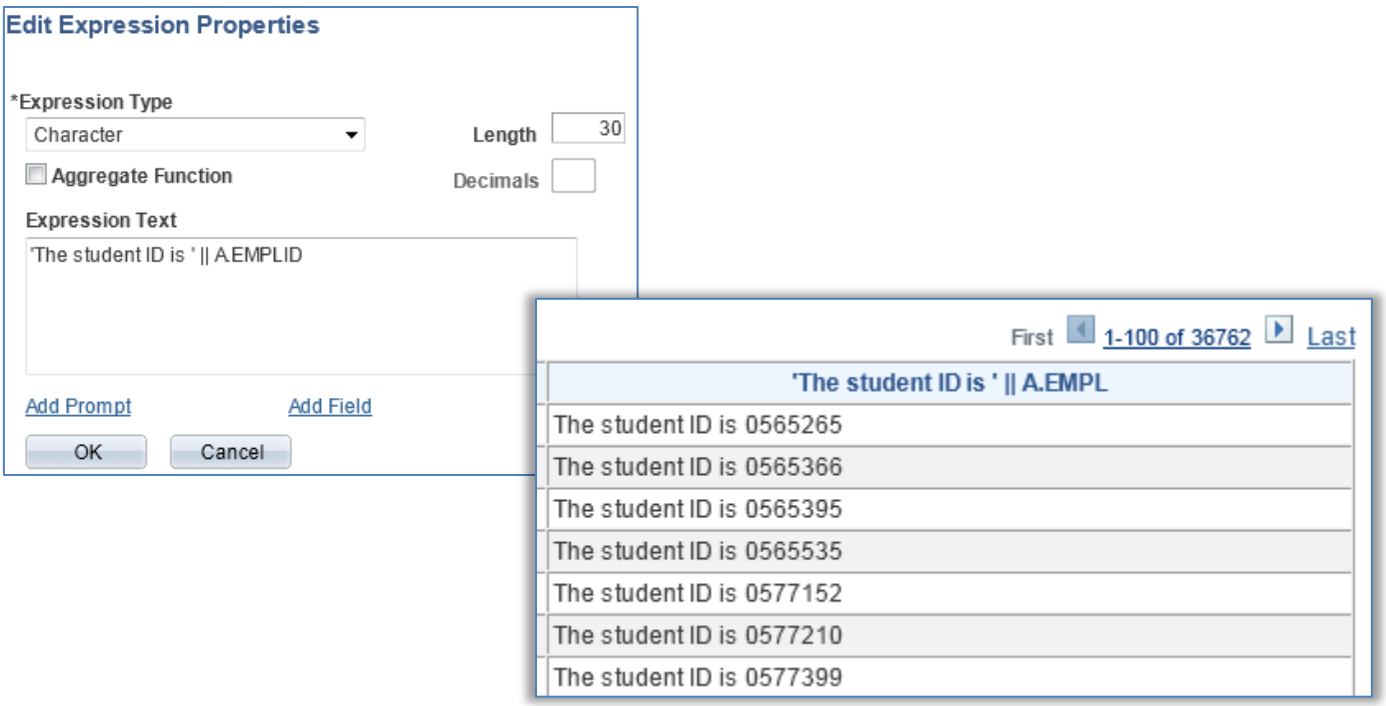
The only operator that applies to strings is concatenation, which is appending one string to the end of another string. The concatenation operator is the double pipe, ||. (The pipe character can usually be entered by holding down the Shift key while pressing the backslash \ key, which is normally above the Enter key.) For example, if A.STR1 contains 'ABC' and A.STR2 contains 'DEFG', then A.STR1 || A.STR2 results in 'ABCDEFG'.

There are not operators that perform calculations on two dates. You can add to or subtract from a date. The numeric value is translated into a number of days. If A.START_DT is 5/14/2013 then the expression A.START_DT + 7 results in 5/21/2013. If the numeric value is a decimal amount, then the time will also be affected. If A.START_DT is 5/14/2013 10:00 AM then the expression A.START_DT + 0.5 results in 5/14/2013 10:00 PM.

Expressions can also contain functions, which take some input values and return an output value. Functions operate similar to how they do in mathematics. Functions in expressions are of the form

FUNCTION_NAME(arg1, arg2, …, argN), where FUNCTION_NAME is the name of the function and arg1, arg2, etc. are its arguments, which are the input values. When using a function in an expression, the name must be spelled exactly as expected, the argument list must be enclosed in parentheses, all arguments must be separated by commas, and all required arguments must be given. (There are sometimes optional arguments.)

Functions that can be used in PeopleSoft Query Expressions are based on the version of the database platform the PeopleSoft application runs on. Currently UHS runs PeopleSoft on Oracle 11g.



'Expressions' Page Objects:

Page Object	*How Used*
Expression Type	Identifies how the resulting data output should be read.
Length	For character and numeric expression types, enter the total number of digits the expression should have. For Number and Signed Number, expression types, the Length field defines the total length of the number (integer portion + decimals portion). For example, if Length = 10 and Decimals = 3, then this means that the integer portion = 7 (Length - Decimals = Integer).
Decimals	Identifies the number of digits that should fall behind the decimal point.
Aggregate Function	If you are entering an aggregate value, such as SUM, AVG, or COUNT, select the Aggregate Function check box. If you put an aggregate function into an expression, you must check the Aggregate Function checkbox, or you will receive the error "not a single-group group function." This is indicating that you are trying to use an

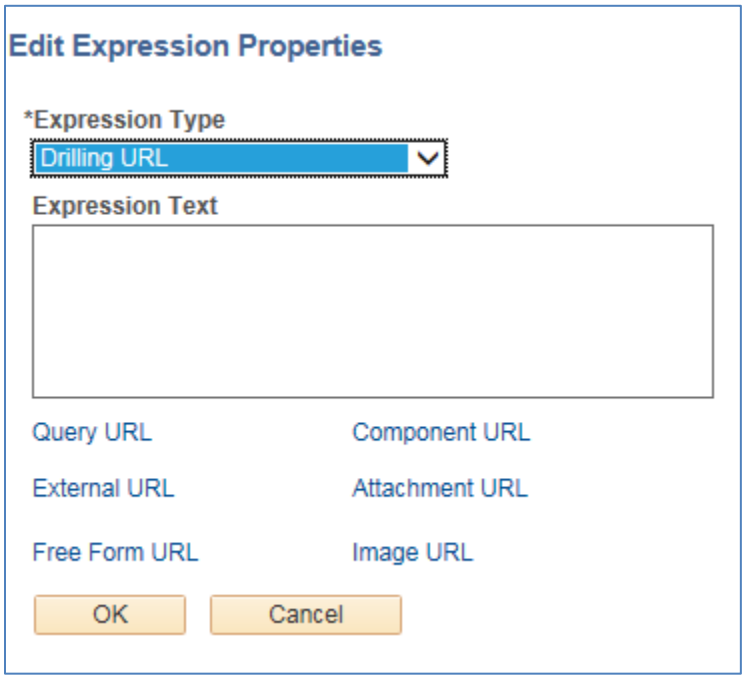
Page Object	*How Used*
	aggregate function on results that are not being aggregated.
Expression Text	In the Expression Text field, enter the expression. Query Manager inserts the expression into the SQL for you..
Add Prompt	Click the Add Prompt button to add prompt properties for this expression.
Add Field	Click the Add Field button to add fields to the expression.
OK	Click the OK button to have Query accept the expression.
Cancel	Click the Cancel button to cancel the current edits made to the epression.

Using Drilling URLs (Special Type of Expression)

Drilling URLs are the URLs that you define by selecting the menu, component, page, portal object, or URL of choice. They are a special type of expression that you can define using the Edit Expression Properties page.

When you build a query using Query Manager, you can define drilling URLs that are associated with this query. These settings are saved into the database, along with prompt, criteria, and so on, as part of the metadata for this query. When you execute this query through Query Manager or Query Viewer, the query results page shows results as links, which you can click to be redirected to a different page in a new browser. Depending on how drilling URLs are defined, the new browser is either a PeopleSoft Pure Internet Architecture page, another query result page, or an external page.

When the “Expression Type” is changed to value “Drilling URL” the below controls appear on the Edit Expression Properties page.



Expression type	To define drilling URLs, you must select the *Drilling URL* option from the Expression type list.
Expression text	Optionally, type the URL in the Expression text text box. **Note:** You have two options: type the URL in this Expression Text box or click the Query URL, Component URL, or External URL links to allow the appropriate system building URLs. If you type the URL directly into the Expression text box, the system does not validate against a value for the correct format.
Query URL	Click to access the Query URL definition widget, where you can select a query to build URLs in a query URL format.
Component URL	Click to access the Component URL definition widget, where you can select a component to build URLs in a component URL format.
External URL	Click to access the External URL definition widget, where you can enter external URL to build URLs in an external URL format.
Attachment URL	Click to access the Attachment URL definition widget, where you can enter attachment URL to build URLs in an attachment URL format.
Free Form URL	Click to access the Free Form URL definition widget, where you can manually enter URLs that will be used to build drilling URLs in report results.

Note: Drilling URLs are saved into database as an expression, so you can access the Expressions page and add the defined drilling URLs as fields. However, you cannot add drilling URLs as criteria because drilling URLs are special type of expressions. You can add drilling URLs as query fields or columns, and then on the query results page, values in that fields or columns will be expanded to fully qualified URLs, which you can click to either run query, access a PeopleSoft Pure Internet Architecture page, or access an external URL.

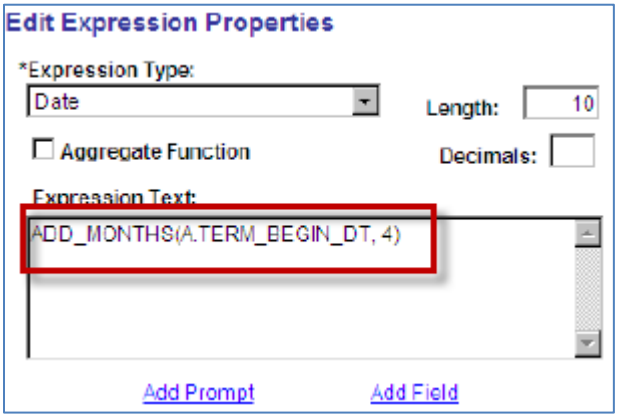
Various steps are involved in building each of the different URL widgets. For detailed steps involved in building each type of widget users can reference site http://docs.oracle.com/cd/E41633_01/pt853pbh1/eng/pt/tpsq/task_DrillingURLinPeopleSoftQuery-0f7f1b.html .

Using Dates in Expressions

Special considerations must be taken when using date fields in expressions. Query Manager automatically converts all values from date fields to text in its processing. For instance, if A.TERM_BEGIN_DT contains the date May 14, 2013, Query Manager turns this into the string "05/14/2013". This can be confirmed by looking at the SQL generated for the query. Consider a query containing a date field, such as the following example; the query made by Query Manager is:

```
SELECT A.STRM, TO_CHAR(A.TERM_BEGIN_DT,'YYYY-MM-DD')
FROM PS_TERM_TBL A
WHERE A.ACAD_CAREER = 'UGRD'
```

In the SELECT clause, the value in the TERM_BEGIN_DT field is passed to the TO_CHAR function. Note also that it is converted to a format such that the components appear in the order year, month, and day (ex. “2009-08-04”). This is then converted again when the query results are displayed.
The consequence of the date being converted to a string is that functions that expect a date value will not work properly. This is because the query engine will not know how to derive a valid date from the string. For example, consider a modified version of the above query to which an expression is added; this expression finds the date four months after the beginning of the term.



Upon running the revised query, this error is reported: “ORA-01861: literal does not match format string (50,380)”. This error is shown when a string to be converted or displayed in a certain manner is incompatible with the instructions of how to convert or display that string; these instructions are the “format string” referred to in the message. Viewing the SQL generated for the query can help show why the query fails:

```
SELECT A.STRM, TO_CHAR(A.TERM_BEGIN_DT,'YYYY-MM-DD'),
ADD_MONTHS( TO_CHAR(A.TERM_BEGIN_DT,'YYYY-MM-DD'), 4)
FROM PS_TERM_TBL A
WHERE A.ACAD_CAREER = 'UGRD'
```

The TO_CHAR function marked in red converts the date to a string. If A.TERM_BEGIN_DT contains the date January 11, 2010, TO_CHAR(A.TERM_BEGIN_DT, 'YYYY-MM-DD') produces the string “2010-01-11”. Next, ADD_MONTHS, marked in blue, attempts to add four months to the first value provided to it. However, the first parameter is receiving text instead of a date; in this instance, the function call is effectively ADD_MONTHS('2010-01-11', 4). The string “2010-01-11” cannot be automatically converted to a date, so this results in an error.

To work around this aspect of Query Manager, the date field that is implicitly converted to a string must be converted back to a date. This can be done by using the TO_CHAR function to transform the string into a date. The following table lists the various types of date fields and the conversion that must be performed to use them in an expression.

Field Type	Convert the Query Manager-converted date to an actual date
Date	TO_DATE(date field, 'YYYY-MM-DD')
Datetime	TO_DATE (datetime field,'YYYY-MM-DD-HH24.MI.SS. ' || CHR(34) || '000000' || CHR(34))
Time	TO_DATE(time field, 'HH24:MI:SS.' || CHR(34) || '000000' || CHR(34))

The ADD_MONTHS expression above would have to be the following in order to work properly in Query Manager: **ADD_MONTHS(TO_DATE(A.TERM_BEGIN_DT, 'YYYY-MM-DD'), 4)**. This

will be converted by Query Manager to ADD_MONTHS(TO_DATE(TO_CHAR(A.TERM_BEGIN_DT,'YYYY-MM-DD'), 'YYYY-MM-DD'), 4) when the SQL is built; the string produced by TO_CHAR will be converted to a date by TO_DATE before being given to ADD_MONTHS, satisfying the requirement that ADD_MONTHS receives a date.

Note that the TO_DATE function will produce actual date values and the expressions that use them will *not* be reformatted by Query Manager. Instead, they will be displayed in Oracle's native format, which is a two-digit year, a three-letter month abbreviation, and a two-digit day, separated by dashes (ex. "24-AUG-09"). To display such a date in the same format Query Manager normally uses, the result of the expression must be converted into a string. The following table shows how TO_CHAR can be used to convert the whole expression to a string that matches Query Manager's display format.

Field Type	Convert date expression to string having Query Manager display format	Minimum Length
Date	TO_CHAR(expression, 'MM/DD/YYYY')	10
Datetime	TO_CHAR(expression, 'MM/DD/YYYY fmHH12fm:MI:SSAM') Note: this format does not exactly match that of Query Manager; the hour is not preceded by an extra space if the hour is less than 10. If the "fm" format modifiers are removed, the format will have the correct length, but the hour will be zero-padded, so the format will still not match exactly. The format model used above is as close to the Query Manager display format as possible without resorting to extremely elaborate conversions.	21
Time	TO_CHAR(expression, 'fmHH12fm:MI:SS.' || CHR(34) || '000000' || CHR(34) || 'AM')	17

In summary, when using dates in expressions:

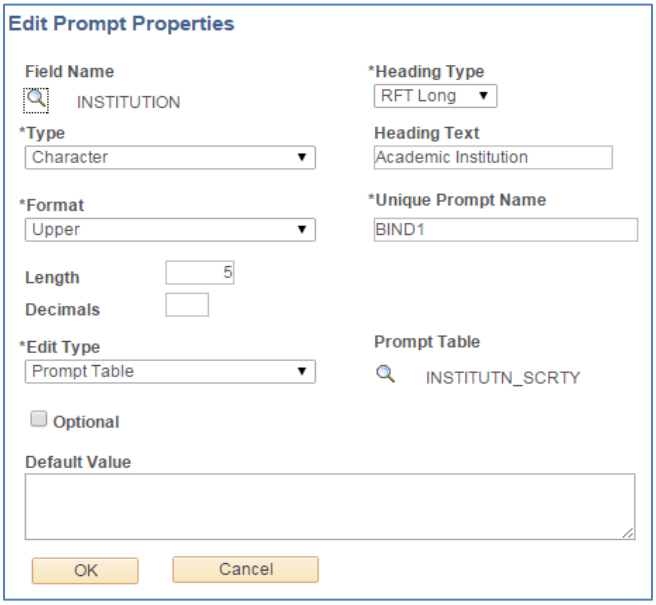
- Query Manager automatically converts the date to a string. You must convert it back to a date using the TO_DATE function.
- Perform other desired operations on the date, such as date arithmetic and using functions such as ADD_MONTHS.
- If the expression will be used as a field, pass the whole expression as the first argument to TO_CHAR and the desired format model as the second argument. To match the default Query Manager display format, consult the above table.
- If the expression will be used as a field and in criteria, make two expressions, one that just converts the string to a date, and one that converts the string to a date then back to a string of the desired format.

Note: This repeated conversion between dates and strings is only needed to work around the automatic conversion of date fields to strings in Query Manager. This is not required when running a SQL statement directly.

'Prompts' Page Tab

The Prompts page allows the user to attach runtime prompts to the query. Adding a prompt lets you further refine a query at the time of running it. To access the Edit Prompt Properties click the Add Prompt button or the Edit button on the Prompts page.

Note. When using a prompt table on a field from a record definition with multiple keys, you must prompt for all higher-level keys before lower-level keys. PeopleSoft Query needs values for the higher-level keys to generate the correct prompt list. Because of this complication, you should minimize using multikey prompt tables.



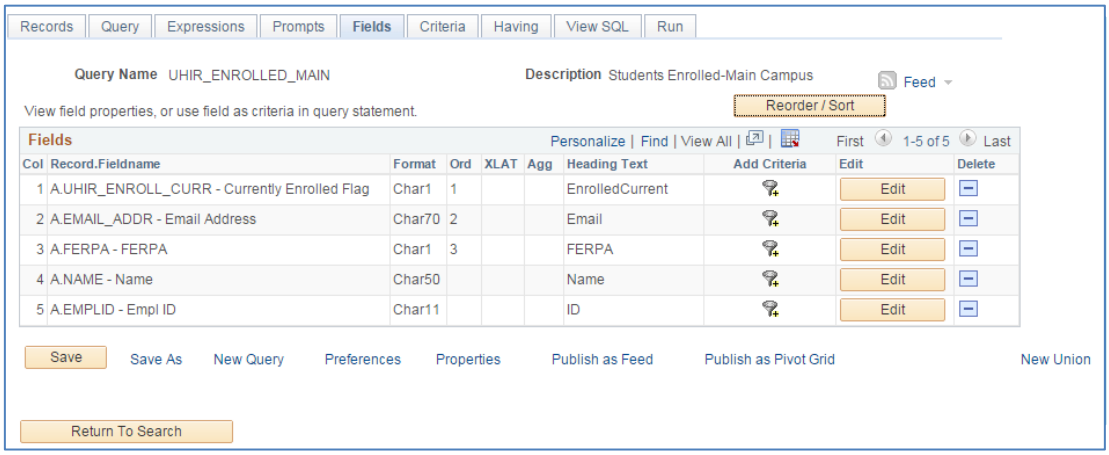
'Prompts' Page Objects:

Page Object	*How Used*
Field	Click the Look Up button next to the Field Name field to select a prompt field. After you select a prompt field, the name of the field appears. PeopleSoft Query looks to the record definition for information about this field and completes the rest of the page based on its properties.
Type	Indicates the type of the field.
Format	Specifies the field format. Over a dozen formats are available, including Name, Phone, Social Security Number, and Zip Code.
Length	Indicates the field length.
Decimals	Defines the number of decimals that are allowed.
Edit Type	Defines the type of field edit for the specified field. No Table Edit is the default value. In general, you should use the same edit type that is used in the field record definition so that this edit type is consistent throughout
Heading Type	Select a heading type for the prompt from the following values: Text: The prompt heading is the free text that you have entered in the text box. RFT Short: The prompt heading is the short name from the record definition. RFT Long: The prompt heading is the long name from the record definition.
Heading Text	Displays the label for the text box where you enter the comparison value. To change the text, select Text from the Heading Type drop-down list box, and then enter the new label in the Heading Text text box.
Unique Prompt Name	A default value that Query Manager generates for globalization. Only baselanguage users can set this value to uniquely identify a query prompt

Page Object	*How Used*
	parameter
Prompt Table	If the edit type is Prompt Table, you can select a prompt table to use. If the edit type is Translate Table, the value in the drop-down list box determines the values used. PeopleSoft Query assumes that the specified field has translate table values associated with it, and that the field is identified as a translate table field in its record definition
Optional	Use this option to define whether the prompt is optional or required. • Select this option to indicate that the query prompt is optional. When the prompt is set to optional, a query may return a large result set because results are not limited or restricted by a prompt value. • Clear this option to indicate that the query prompt is required. By default, the Optional option is cleared and a valid prompt value must be selected or entered when you run the queries that have prompts. **Note:** A required prompt value must be validated based on the Edit Type list. Prompt with the Edit Type set to Yes/No Table cannot be set as optional prompt because there are only two valid values, Y and N.
Default Value	Use this field to set a default value for the prompt. The value in this field is used to populate the prompt when no other prompt value is selected or entered. For character fields, you can add a default prompt value that has up to 254 characters long. **Note:** Entering a default value for a prompt set the prompt to required. If the Optional option is selected, when you enter a default prompt value for the prompt, the Optional option is cleared and a message appears saying: "Optional prompt cannot have default value. Prompt is now set to be required." Default prompt value is not a translatable field. Tree prompts cannot be used as the default or optional prompt values.

'Fields' Page Tab

The Fields page allows the user to edit the fields selected on the Query. This page lists the various fields that will be displayed as output when the query is run. The user can establish column order, change headings, create translate values, change output order, and summarize data through aggegation.

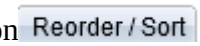


.

'Fields' Page Objects:

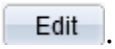
Page Object	*How Used*
Reorder/Sort	Click this button to display the Edit Field Ordering page, which enables you to change the column order and/or sort order for multiple fields.
Col	Displays the chronological sequence that the fields will be displayed as columns of output
Record.Fieldname	Indicates the record alias and the field name in the database
Format	Field type and length for each field listed in the output.
Ord *(order)*	Shows one or more fields selected to sort your query output. If the field is the first sort field, a 1 appears in the cell and the system sorts rows that are based on this field first. The second sort field that is selected is numbered 2, and so on. You can also specify a descending sort order. The letter D appears if you are sorting fields in descending order.
XLAT *(translate)*	Indicates whether the field has a translate value attached to it (N-*none*, S-*short*, L-*long*); it directs how the text will appear in the output for that field. The table you're querying may include fields that use the Translate table. If so, the field itself contains a short code for which the Translate table provides a set of corresponding values. For example, if the table includes an EFF_STATUS field, the short code value is 'A' or 'I', which the Translate table translates into long text 'Active' and 'Inactive'. If a field has values on the Translate table, a letter appears in the XLAT column for that field In your query results, you might want to display the translated value rather than the code (for example, 'Active' instead of 'A'). To instruct PeopleSoft Query to make this substitution, specify L as the translate value. Translate tables are effective-dated, so you must select which effective date to use for them.

Page Object	*How Used*
Agg *(aggregate)*	Indicates that the field has an aggregate function (Sum, Count, Min, Max, Average) applied to it.
Heading Text	The output heading label assigned to appear at the top of the column in the query output for each field listed.
 **Add Criteria**	Click the Use as Criteria /Add Criteria button to open the Edit Criteria Properties page, where you can determine how this field will be used as a criterion for the current query.
Edit	Click this button to display the Edit Field Properties page and edit field characteristics.
 **Delete**	Click the Delete button to delete the field from the query.

To change column and row sort ordering, click on the Reorder/Sort button  .

'Edit Field Ordering' Subpage Objects:

Page Object	*How Used*
New Column	Enter the new column number to reorder the columns. Columns that are left blank or assigned a zero are automatically assigned a number.
Column	Current column order.
Record.Fieldname	Indicates the record alias and the field name in the database
Order By	Current sort order.
Descending	Select this check box to sort fields in descending order.
New Order By	Enter the new sort order number to change the sort order. Enter zero to remove a sort order. If the field is the first sort field, enter 1, and the system sorts rows based on this field first. To designate the second sort field, enter 2, and so on.

To change field properties, click on the Edit button  .

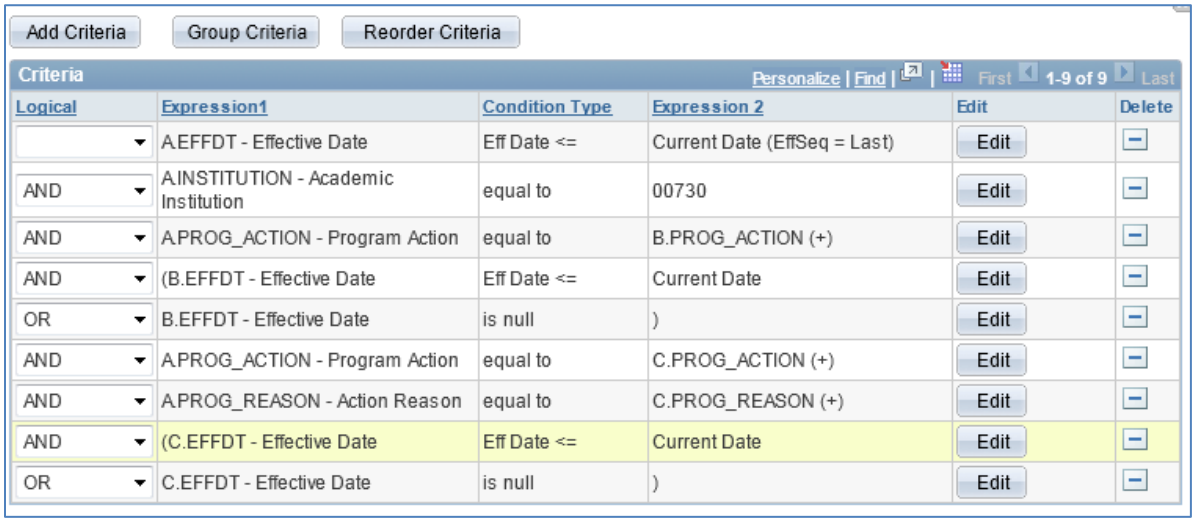
'Edit Field Properties' Subpage Objects:

Page Object	*How Used*
Field Name	The name of the field for which you are editing properties.
Heading	Choose a column heading from the following: ***No Heading*** – The column does not have a heading. ***Text*** – The column heading is the text that you have entered in the text box. ***RFT Short*** – The column heading is the short name from the record definition. ***RFT Long*** – The column heading is the long name from the record definition.

Page Object	*How Used*
Unique Field Name	Used to uniquely identify this field when generating the background SQL statement. **There is no need to change the default value**, which is a single-letter alias for the record followed by the record field name (for example A.NAME or B.EMPLID). **It is strongly recommended that this value not be changed from the default**.
Aggregate	If you are using aggregate values, select the aggregate function value for this field. An aggregate function is a special type of operator that returns a single value based on multiple rows of data. When your query includes one or more aggregate functions, PeopleSoft Query collects related rows and displays a single row that summarizes their contents. ***None:*** *W*ill not use aggregate functions. ***Sum:*** Adds the values from each row and displays the total. ***Count:*** Counts the number of rows. ***Min (minimum):*** Checks the value from each row and returns the lowest one. ***Max (maximum):*** Checks the value from each row and returns the highest one. ***Average:*** Adds the values from each row and divides the result by the number of rows.

'Criteria' Page Tab

You define selection criteria to selectively retrieve the data that you want. Selection criteria refine your query by specifying conditions that the retrieved data must meet. Criteria can be added from the Criteria page by clicking on the page's Add Criteria button, and by clicking on the Add Criteria funnel icon from the Query page, Expressions page, and Fields page.



Add Criteria | Group Criteria | Reorder Criteria

Criteria — Personalize | Find | First 1-9 of 9 Last

Logical	Expression1	Condition Type	Expression 2	Edit	Delete
	A.EFFDT - Effective Date	Eff Date <=	Current Date (EffSeq = Last)	Edit	
AND	A.INSTITUTION - Academic Institution	equal to	00730	Edit	
AND	A.PROG_ACTION - Program Action	equal to	B.PROG_ACTION (+)	Edit	
AND	(B.EFFDT - Effective Date	Eff Date <=	Current Date	Edit	
OR	B.EFFDT - Effective Date	is null	)	Edit	
AND	A.PROG_ACTION - Program Action	equal to	C.PROG_ACTION (+)	Edit	
AND	A.PROG_REASON - Action Reason	equal to	C.PROG_REASON (+)	Edit	
AND	(C.EFFDT - Effective Date	Eff Date <=	Current Date	Edit	
OR	C.EFFDT - Effective Date	is null	)	Edit	

'Criteria' Page Objects:

Page Object	*How Used*
Logical	Represents how the criteria rows will be compared with each other; it contains information regarding how the various criteria should be applied in relation to one another. Any rows after the first row must include either an AND or OR logical value in the Logical column to specify whether you want the rows to meet this criterion in addition to other criteria that you have defined or as an alternative criterion. The first criterion that you define does not have a value in this column. The default for subsequent criteria is AND. On some occasions, you will need to write a query that returns rows that meet some criteria, but not all of them. For instance, you may be asked to limit the results to those for which the program status is either 'AC' or 'LA'. This could be implemented using the criterion "PROG_STATUS in list 'AC', 'LA'", but if the criteria involve multiple fields or cannot be enumerated in a list, you will need to create multiple criteria and link them with the OR operator.
Expression I	Used to specify what is being compared in that criteria row. Right-click for options of Field or Expression. From there a dialog will be presented to select the field or to create an Expression. Select the Field option if you want to base the selection criterion on another field's value, usually a field in another record component. To compare the

Page Object	How Used
	values from fields in two records, you must join the record components. When you select this option, you must then select a condition type. Select the Expression option if you want PeopleSoft Query to evaluate an expression that you enter before comparing the result to the value in the selected field.
Condition Type	States how the Expression 1 is to be compared with Expression 2.
Expression 2	Use as the comparison value against Expression 1.
Edit	Allows editing of row criteria parameters.
Delete	Deletes the criteria row.
Add Criteria	Adds a new criteria row.
Group Criteria	Groups multiple criteria rows to facilitate execution precedence.
Reorder Criteria	To reorder the criteria for your query, click the Reorder Criteria button, enter the new positions for the criteria on the Edit Criteria Ordering page, and click OK. You can also click the Add Criteria button from this Criteria page to add additional criteria, and you can click the Group Criteria button to group your criteria logically.

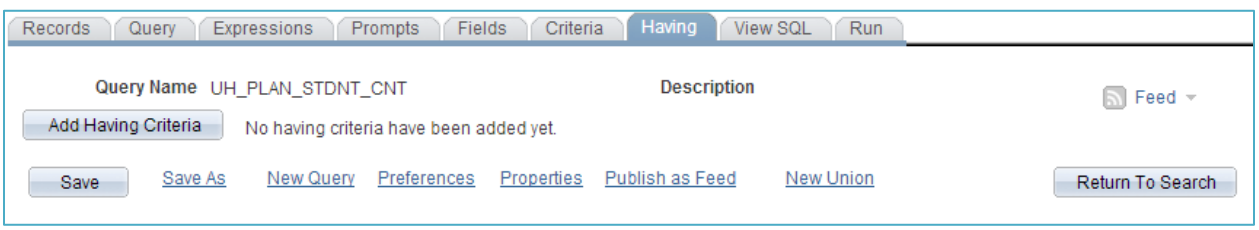
'Having' Page Tab

The "Having" page allows the user to create and edit criteria for fields that have aggregate functions associated with them. You cannot use fields that are being aggregated as regular criteria. SQL does not support the use of aggregate functions in WHERE clauses. Therefore, after you have applied an aggregate function to a field, you cannot use that field in your selection criteria, which corresponds to a SQL WHERE clause.

When you want to filter rows based on the results of an aggregate function, Query Manager enables you to create HAVING criteria. You might use such criteria, for example, when you want a list of the students having two or more degrees. In SQL, a HAVING clause is similar to a WHERE clause for rows of data that have been aggregated into a single row of output. The system evaluates WHERE clauses by looking at the individual table rows before they are grouped by the aggregate function, and then it evaluates HAVING clauses after applying the function.

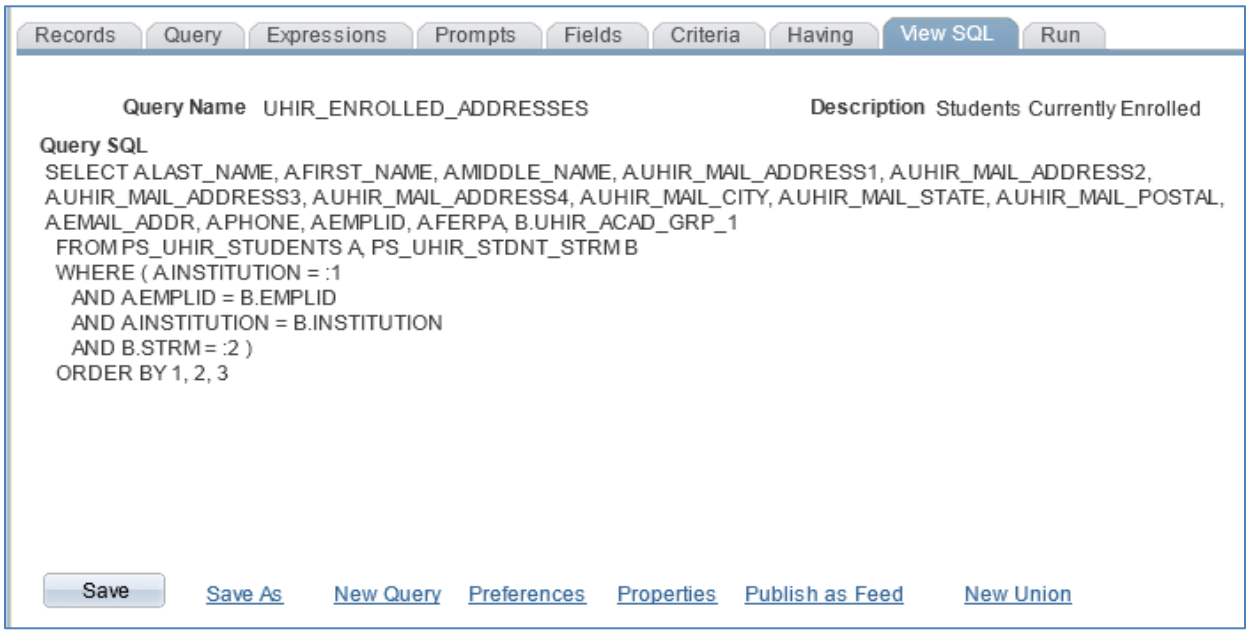
When you click the Add Criteria icon from the Fields or Query pages for an aggregate field, new criteria is added to the Having page instead of the Criteria page. Add selection criteria using the Having page in the same way that you add selection criteria using the Criteria page. Keep in mind that PeopleSoft Query compares the result of applying the aggregate function to the comparison value.

.



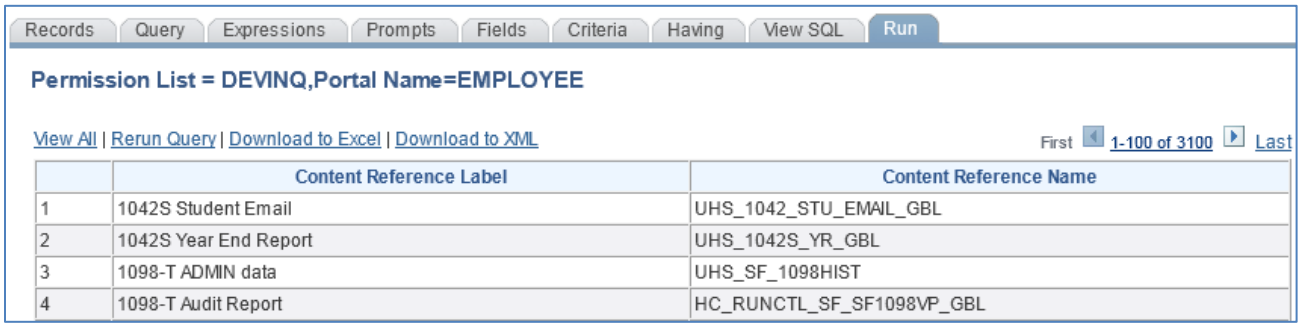
'View SQL' Page

The 'View SQL' page displays the underlying SQL (Structured Query Language) code that Query Manager generates based on how you have designed the query. Viewing the SQL statement can be useful in troubleshooting queries that utilize records that have underlying joins associated with Security Search records. This page is *Read-only* and cannot be directly edited however the content can be copied. To copy the SQL statement, highlight the text of the statement and copy it using your browser's copy command. Paste it into another application, if desired.



'Run' Page

The 'Run' page displays the output generated by the query. You can view the results of the query, rerun the query, as well as download the results of the query from this page.



	Content Reference Label	Content Reference Name
1	1042S Student Email	UHS_1042_STU_EMAIL_GBL
2	1042S Year End Report	UHS_1042S_YR_GBL
3	1098-T ADMIN data	UHS_SF_1098HIST
4	1098-T Audit Report	HC_RUNCTL_SF_SF1098VP_GBL

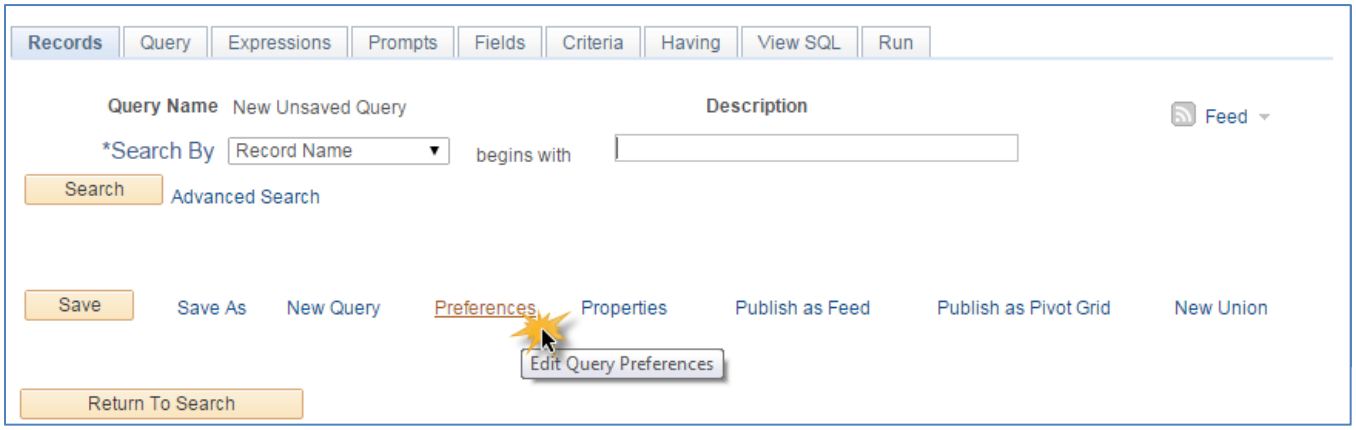
'Run' Page Objects:

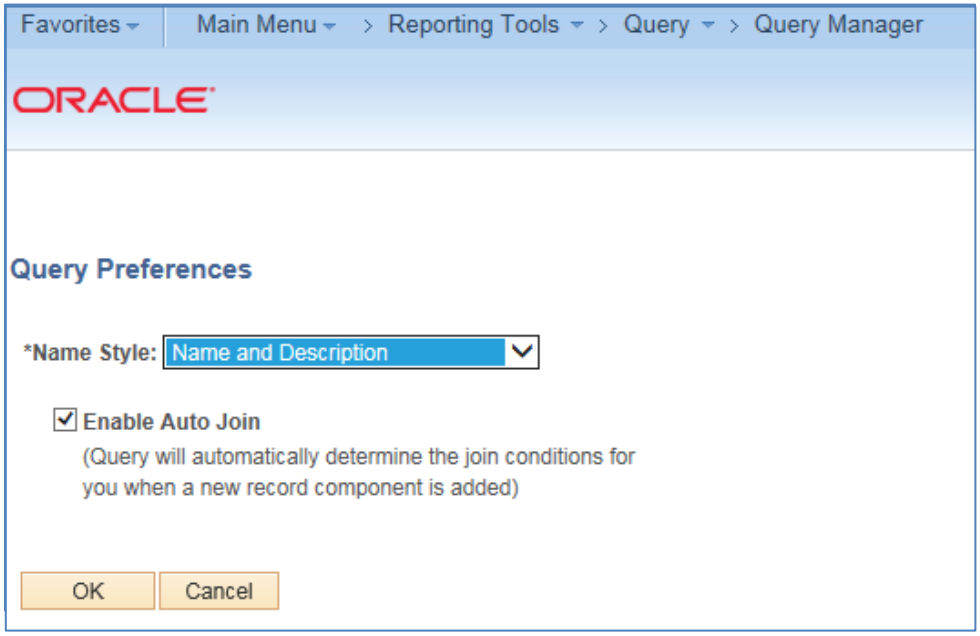
Page Object	*How Used*
View All	Click this link to view all rows and use scroll bar to navigate.
Rerun Query	Click this link to rerun your query preview. If you have made changes to your query since the last preview, you must rerun the query to see the effect

Page Object	*How Used*
	of your changes. Note. If you do not want to rerun the query every time you want to preview it, deactivate the Enable Auto Preview feature. When this feature is active, your query automatically runs each time you select the Run tab. To activate Enable Auto Preview, click the Preferences link and then select Enable Auto Preview. (The Preferences link is available on every page except the Run page.)
Download to Excel	Click this link to download your query to Microsoft Excel.
Download to XML	Click this link to download your query result as XML format to another browser window.
First	Click the First link to jump to the first subset of displayed items when scrolling through the display groups.
and	Show Next Rows/Show Previous Rows scroll bars. Use the left and right arrow scrollbars to go through the rest of the grouped items displayed in a range.
Last	Click the Last link to jump to the last subset of displayed items when scrolling through the display groups.

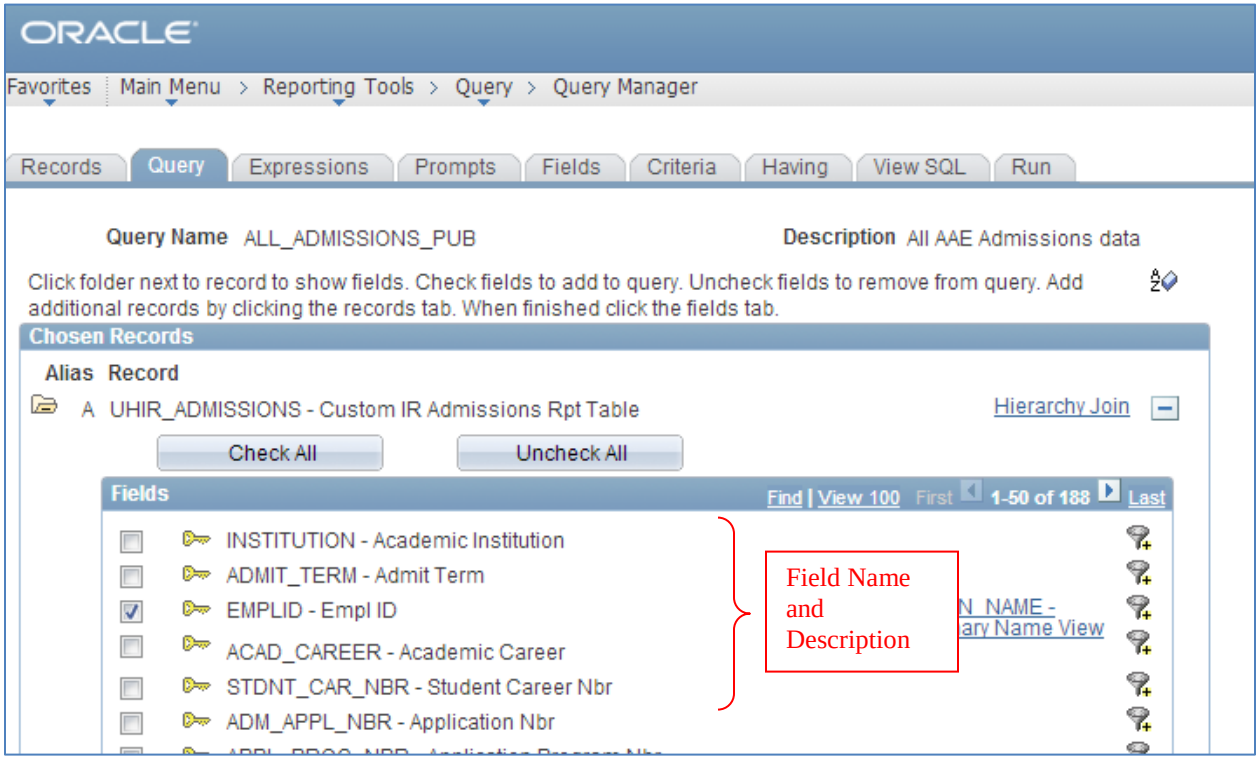
'Query Preferences' Page

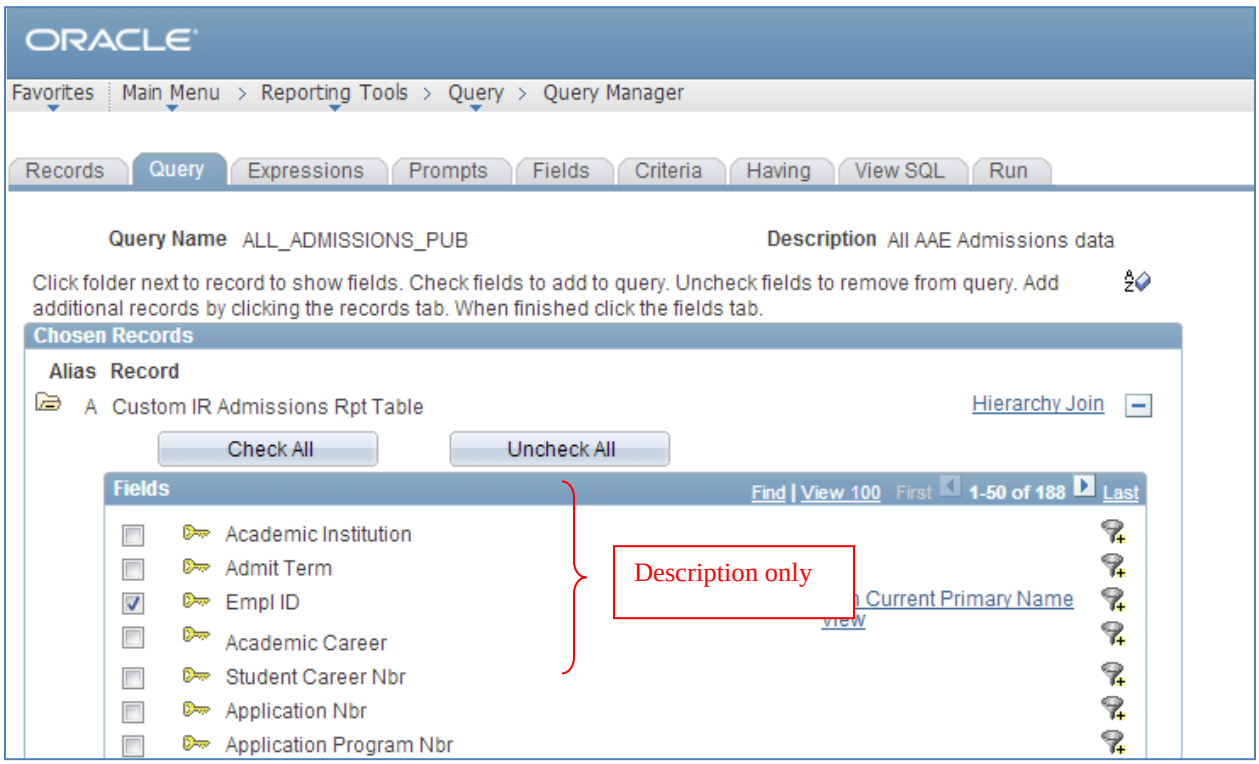
Clicking on the Preferences link navigates you to the 'Query Preferences' page. This page allows you to specify view preferences for record fields, automatic joins, and automatic preview.





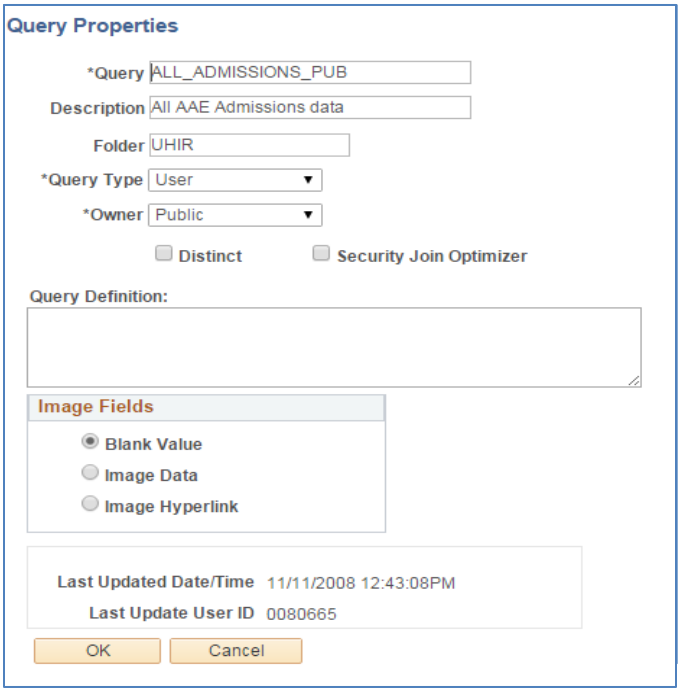
Preference	*How Used*
Name Style	Specify how record and field names are displayed. It is recommended the default selection 'Name and Description' be used. • Description only – list only the descriptions of the field. • Name and Description – list the name of the field and its description. This is the default selection.
Enable Auto Join	Select to indicate that your query should automatically determine the join conditions when a new record component is added. It is recommended to keep this default enabled.


ORACLE
Favorites
Main Menu > Reporting Tools > Query > Query Manager
Records
Query
Expressions
Prompts
Fields
Criteria
Having
View SQL
Run
Query Name ALL_ADMISSIONS_PUB
Description All AAE Admissions data
Click folder next to record to show fields. Check fields to add to query. Uncheck fields to remove from query. Add additional records by clicking the records tab. When finished click the fields tab.
Chosen Records
Alias Record
A UHIR_ADMISSIONS - Custom IR Admissions Rpt Table
Hierarchy Join
Check All
Uncheck All
Fields
Find | View 100 First 1-50 of 188 Last
INSTITUTION - Academic Institution
ADMIT_TERM - Admit Term
EMPLID - Empl ID
ACAD_CAREER - Academic Career
STDNT_CAR_NBR - Student Career Nbr
ADM_APPL_NBR - Application Nbr
Field Name and Description


ORACLE
Favorites
Main Menu > Reporting Tools > Query > Query Manager
Records
Query
Expressions
Prompts
Fields
Criteria
Having
View SQL
Run
Query Name ALL_ADMISSIONS_PUB
Description All AAE Admissions data
Click folder next to record to show fields. Check fields to add to query. Uncheck fields to remove from query. Add additional records by clicking the records tab. When finished click the fields tab.
Chosen Records
Alias Record
A Custom IR Admissions Rpt Table
Hierarchy Join
Check All
Uncheck All
Fields
Find | View 100 First 1-50 of 188 Last
Academic Institution
Admit Term
Empl ID
Academic Career
Student Career Nbr
Application Nbr
Application Program Nbr
Description only

'Query Properties' Page

Clicking on the Properties link navigates you to the 'Query Properties' page. This page allows you to set metadata and properties of the query as a whole.



'Query Properties' Page Objects:

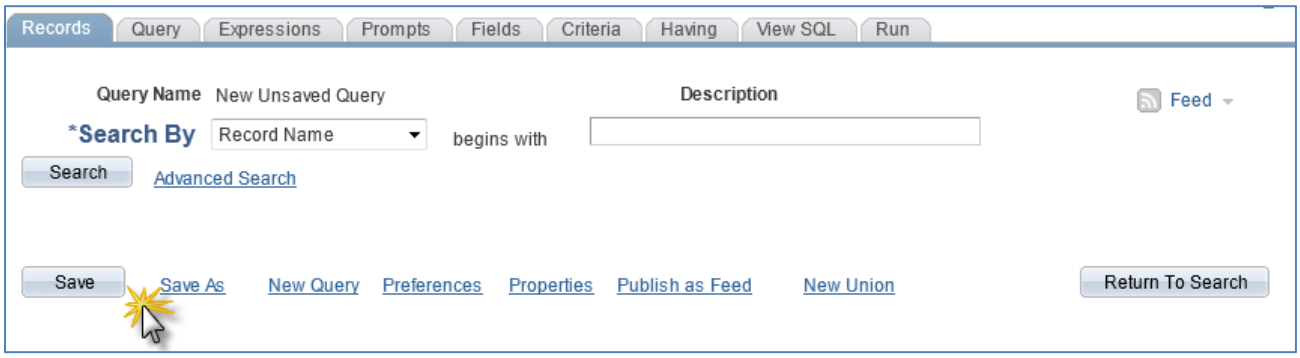
Page Object	*How Used*
Query	Use this field to edit the Name of the Query.
Description	Edit the Description of the query.
Folder	Queries can be grouped into folders. Enter the name of the Folder in which you would like the query to reside. To create a new folder, enter the name of the new folder.
Query Type	Users should leave the query as default type 'User'.

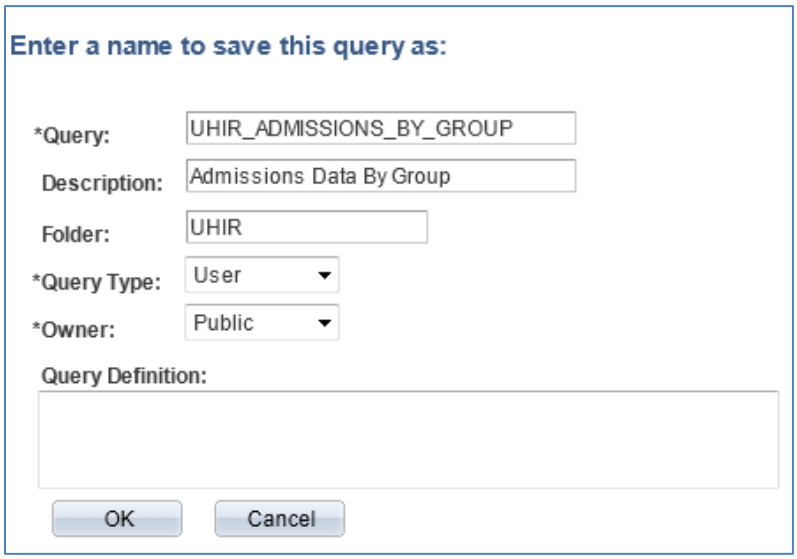
Page Object	*How Used*
Owner	Queries can either be accessible only by a single user id or globally by all users. Private: If you select this option, only the user ID that created the query can open, run, modify, or delete the query. Public: If you select this option, any user with access to the records used by the query can run, modify, or delete the query (if he or she has access to public queries).
Distinct	Query Manager allows for the removal of duplicate rows of data returned by a query. Eliminating duplicate rows of data will return a more accurate view of the information in the database. Select this check box to eliminate duplicate rows in a query result. Some queries return the same row more than once because it satisfies the query in more than one case. For example, if you query the JOB record to return all JOBCODES, you receive multiple rows that look identical because some employees have the same JOBCODE. If you select the Distinct check box, you receive each JOBCODE once. This option is not visible for union selections because unions are automatically distinct.
Security Join Optimizer	If this query contains multiple joins to the same query security record, define whether it should be run optimized. Changing the default value for this feature should only be done by advanced query writers. • Select to enable this query to join once to the first security record. If this is a non self-join (same security tables) or a self-join, the query joins security table once. If this is a non self-join with different security tables, the query joins security table multiple times using the Inner Join logic and the Security Join Optimizer option is not used. • Clear to enable this query to join multiple times to the security record. If this is a non-self join with same security tables or a self join, the query joins security table multiple times using the Inner Join logic. If this is a non-self join with different security tables, the query joins security table multiple times using the Inner Join logic and the Security Join Optimizer option is not used. Note. The concepts of Security Join Optimizer are also applied when you perform Left Outer Joins with security records.
Query Definition	Free text area that you can use to further describe your query. Users should include in the Query Definition historical date/user information and a brief summary of the purpose and changes made to the query.
Last Updated Date/Time	This is a *Read-only* field that stores the date and time of when the query was last edited.
Last Update User ID	This is a *Read-only* field that stores the user who last edited the query.

.

'Save/Save As' Page

Clicking on the Save button or Save As link navigates you to the 'Save Query' page where you can save your query design edits.





'Save/Save As' Page Objects:

Page Object	*How Used*
Query	Use this field to edit the Name of the Query.
Description	Edit the Description of the query.
Folder	Queries can be grouped into folders. Enter the name of the Folder in which you would like the query to reside. To create a new folder, enter the name of the new folder.
Query Type	Users should leave the query as default type 'User'.
Owner	Queries can either be accessible only by a single user id or globally by all users. Private: If you select this option, only the user ID that created the query can

Page Object	*How Used*
	open, run, modify, or delete the query. Public: If you select this option, any user with access to the records used by the query can run, modify, or delete the query (if he or she has access to public queries).
Query Definition	Free text area that you can use to further describe your query. Users should include in the Query Definition historical date/user information and a brief summary of the purpose and changes made to the query.

.

CHAPTER 6 – Basic Query Writing

Overview

In this chapter, you will learn general query methodology and query structural process flow. Some of the basic concepts and features of Query Design will be shown, including how to create a query from one table, manipulate data in a query, and saving a query.

Example

Create a new query that will show the students that were enrolled at the University of Houston – Main on the last day of the '1730' semester.

Creating a Basic Query

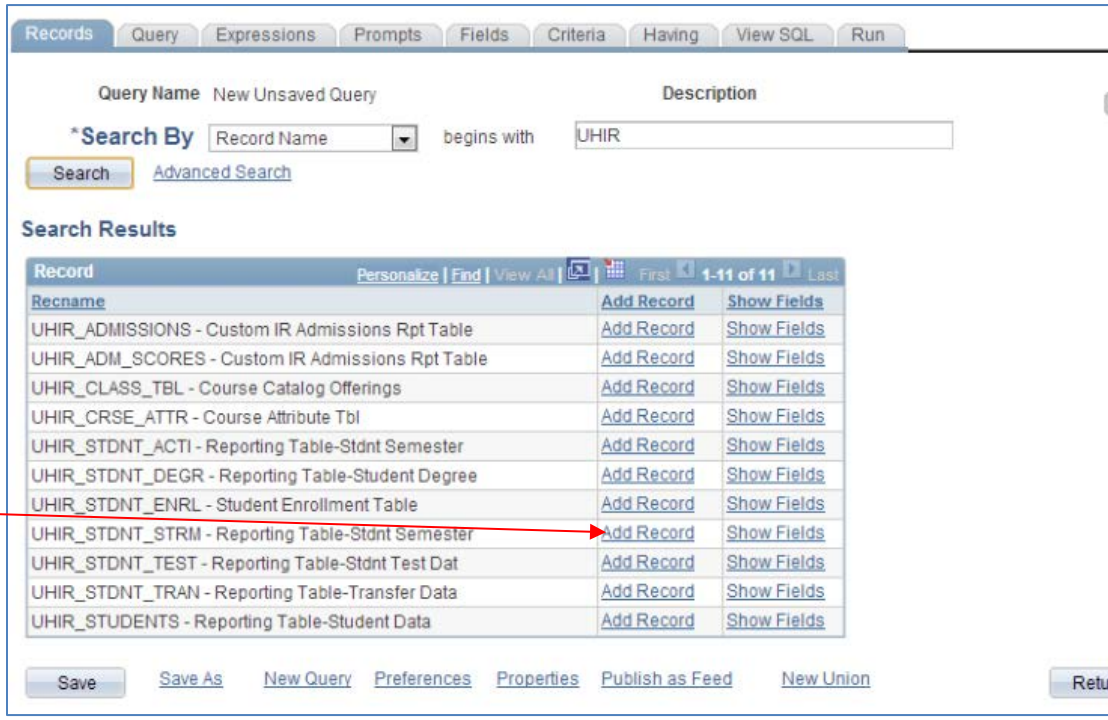
Step 1: Select the Primary Record

The first step in creating a query is selecting a record. The selected record establishes the primary focus of a query. Records are added to the query definition using the '**Records**' page.

(a) Navigate to *Reporting Tools, Query, Query Manager* and Click on *Create New Query* link.

(b) Type **UHIR_** in the "begins with" edit box and click Search.

(c) Select the record to be used as the primary focus of the query by clicking on the *Add Record* link next to the **UHIR_STDNT_STRM** record.



Step 2: Select the Fields on the Report

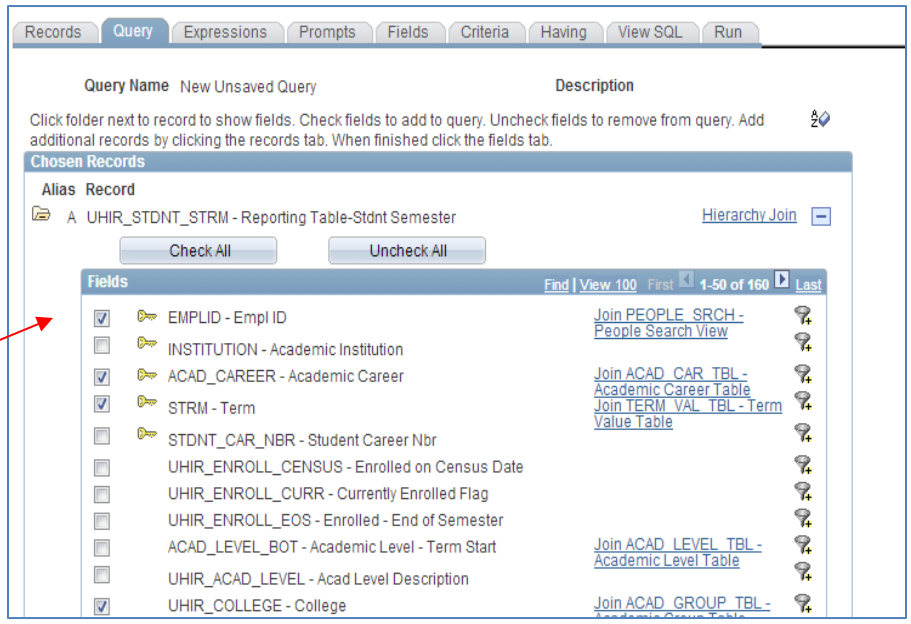
The fields within the primary record added to the query definition are displayed under the '**Query**' page.

(a) Select the fields that you want displayed in the output. A field can be chosen by selecting the check box next to the fieldname/description or by clicking the '**Check All Fields**' button.

Field(s) to Select:

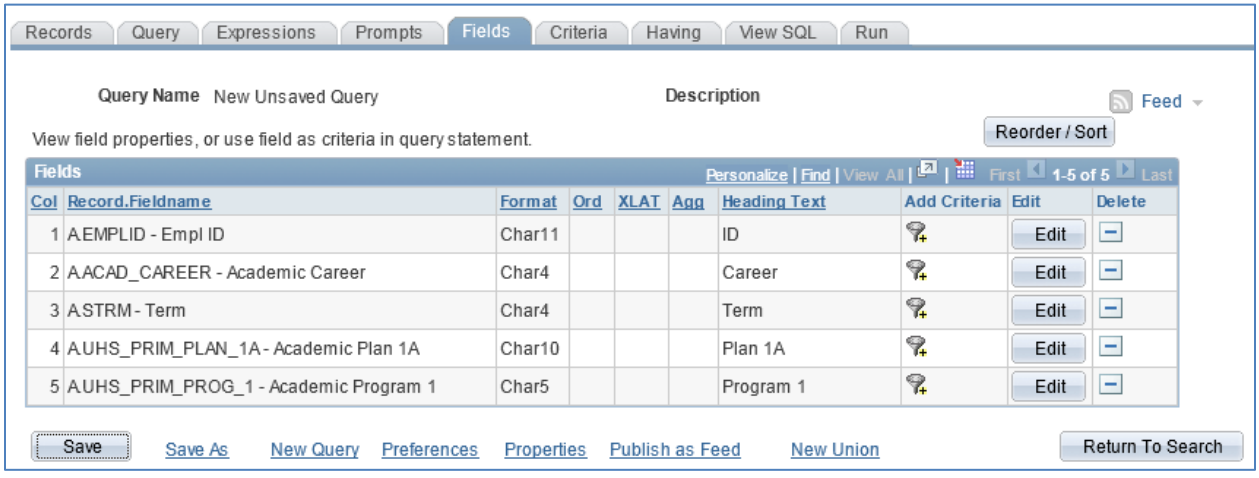
- EMPLID
- ACAD_CAREER
- STRM
- UHS_PRIM_PROG_1
- UHS_PRIM_PLAN_1A

(b) When you have selected the desired fields, select the Fields tab.



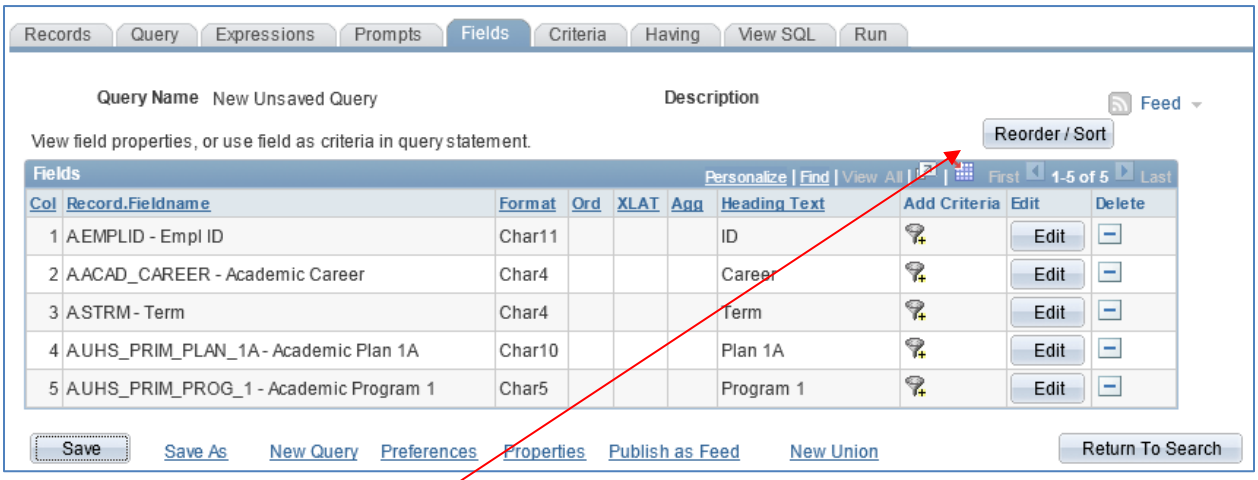
Step 3: Customize Query Field Output

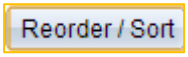
Click on the **Fields** tab after selecting the fields. You will notice that the fields displayed are the fields that were previously selected on the **Query** tab.



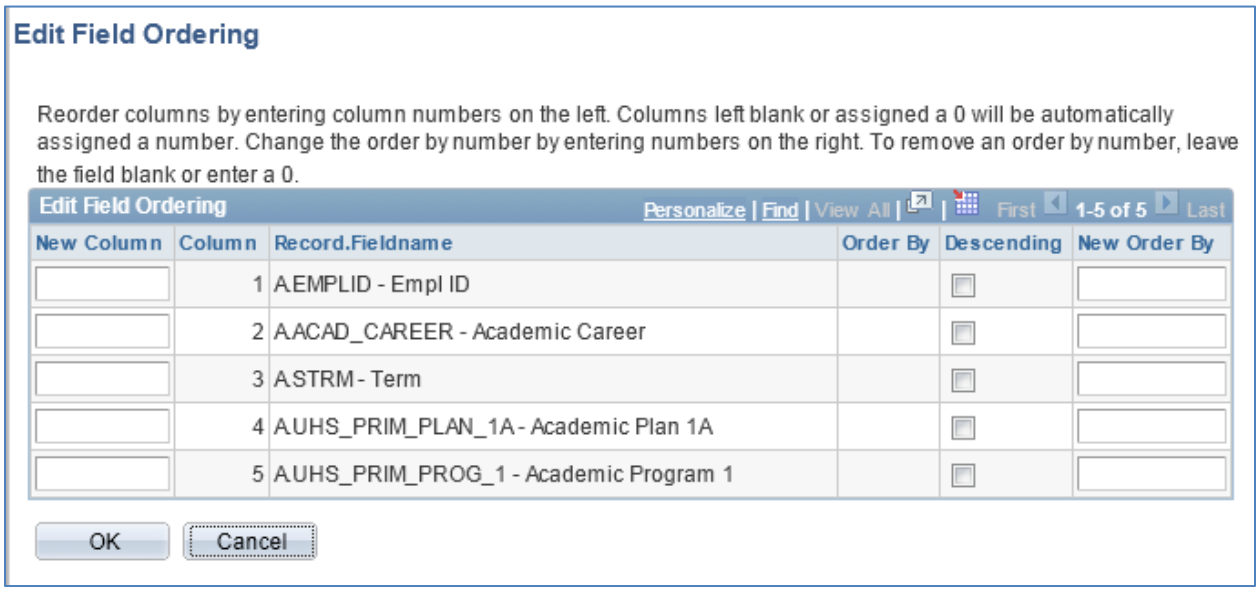
(a) Change the Column Order

The column order is displayed in the column title ***Col***. The order in which fields are selected is the order in which the fields will be displayed in the output/report.



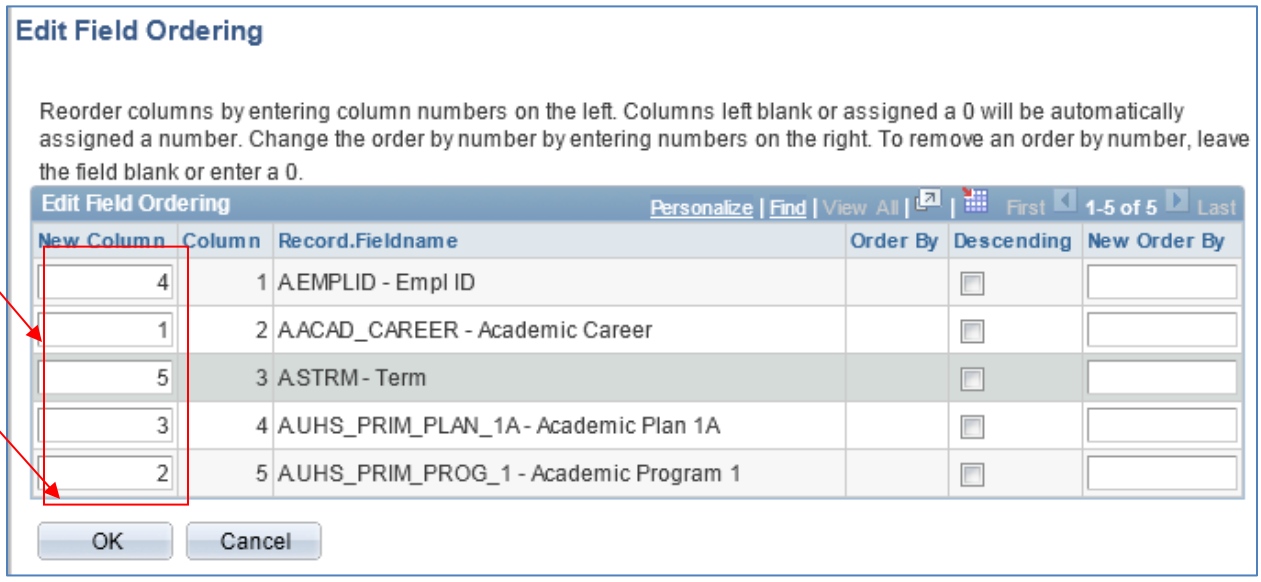
The default column order does not always produce the desired results. Click on the  button to change the column order on the report.

The **Edit Field Ordering** page will be displayed

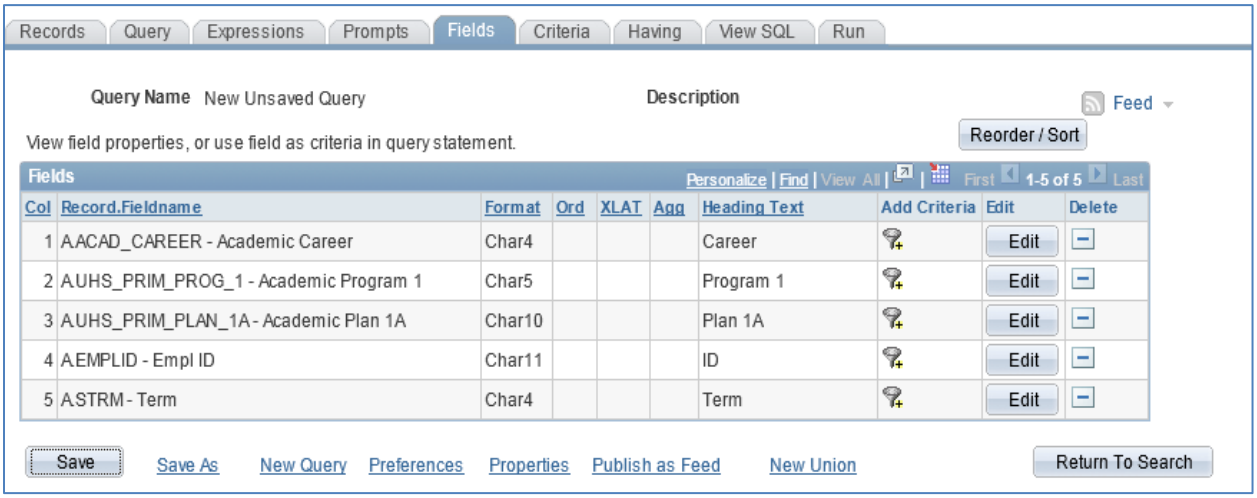


Enter the new column order numbers as indicated.

Click OK

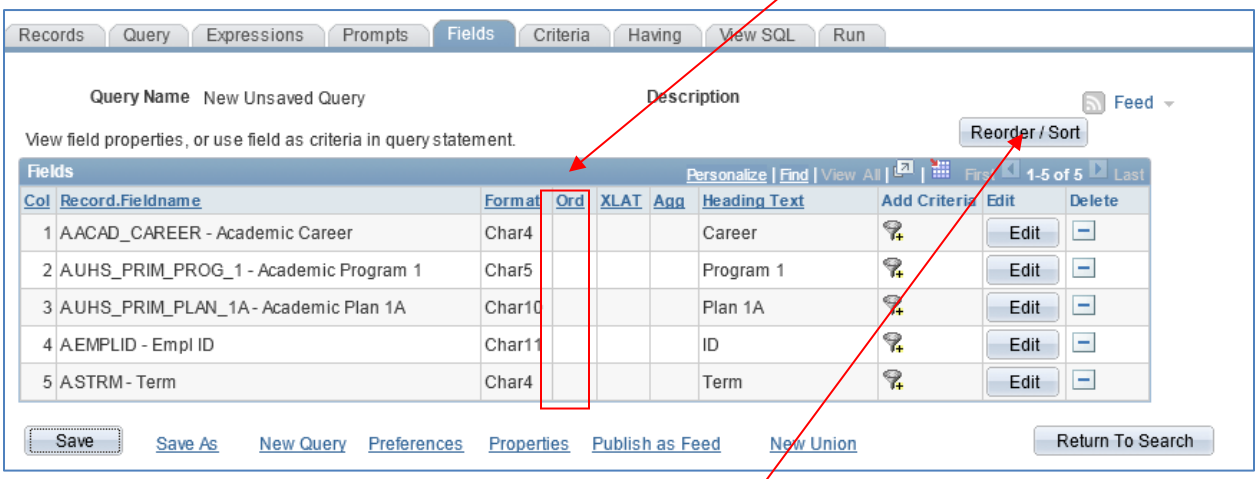


The **Edit Field Column Order** page is redisplayed with the fields in the new column order.



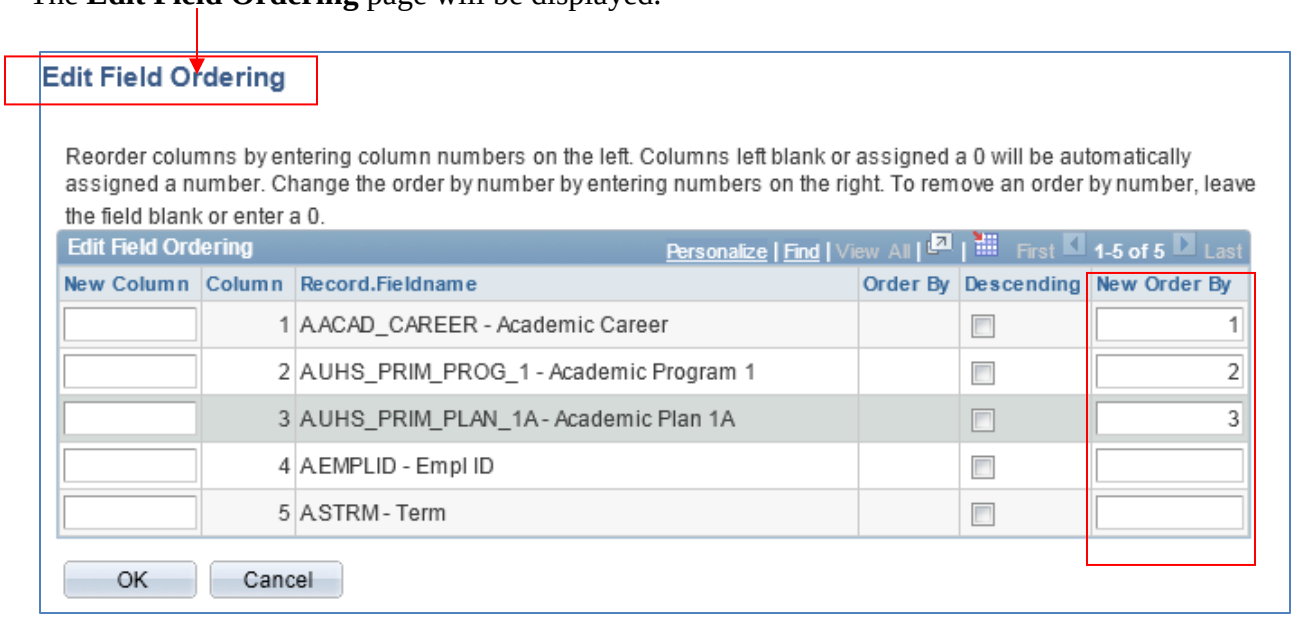
(b) Change the Sort Order

The report sort order is displayed in the column title ***Ord***. If the **Ord** value is blank, then the rows will be displayed in the order the rows were entered in the source records instead..



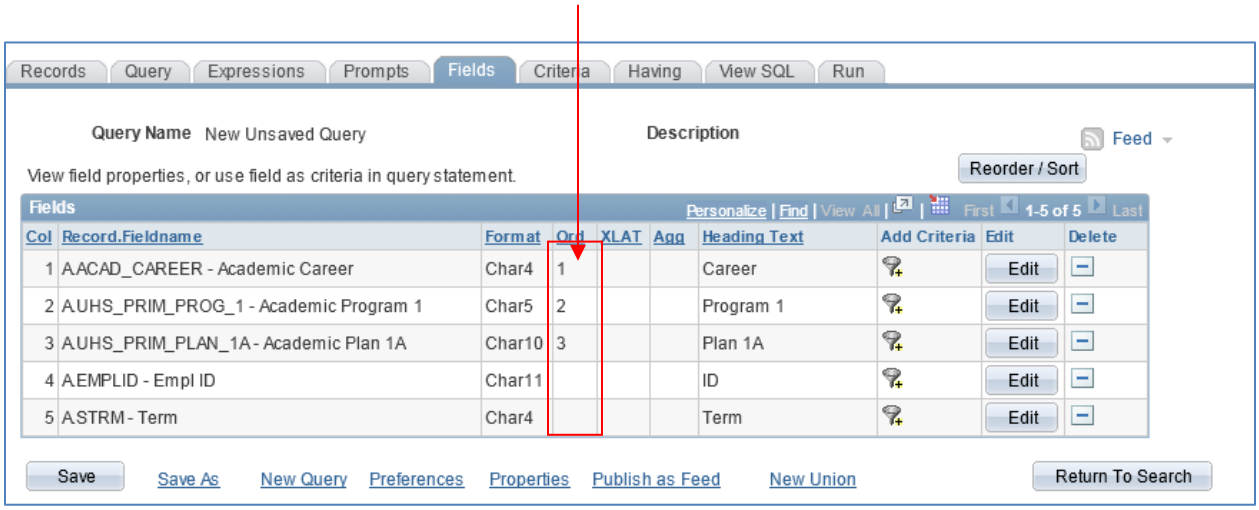
To change the report sort order, click on the Reorder / Sort button.

The **Edit Field Ordering** page will be displayed.



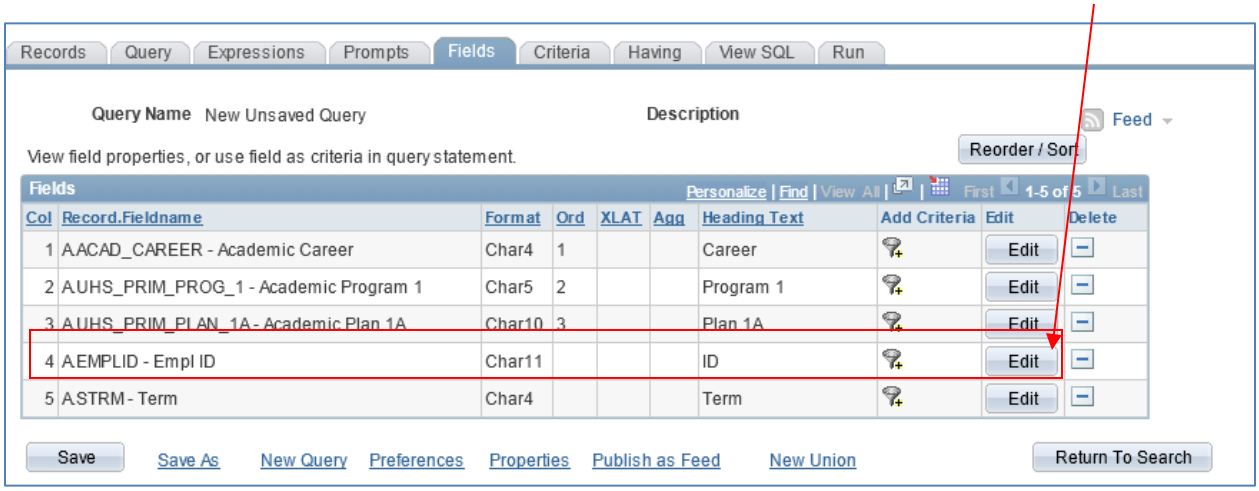
Enter the new field sort order numbers as indicated and click OK.

The updated sort order is displayed in the **Ord** column.



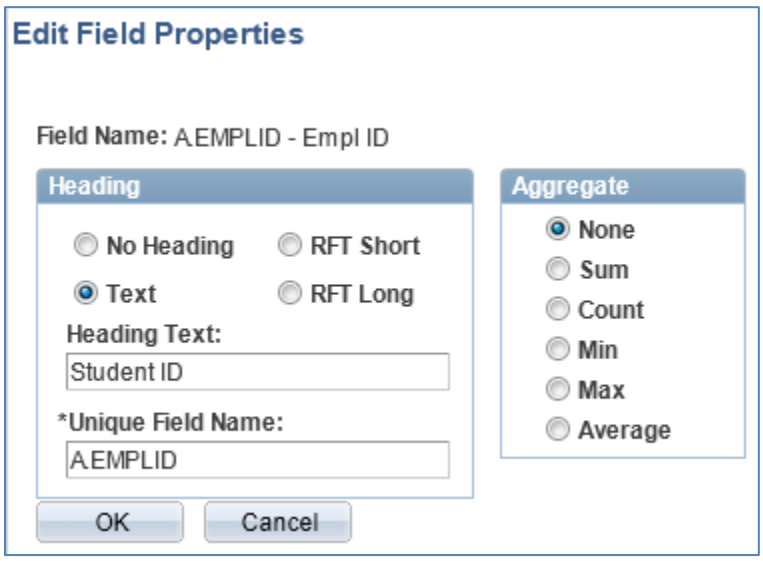
(c) Change Field Properties

Customize field properties such as heading text and aggregation by clicking on the Edit button.

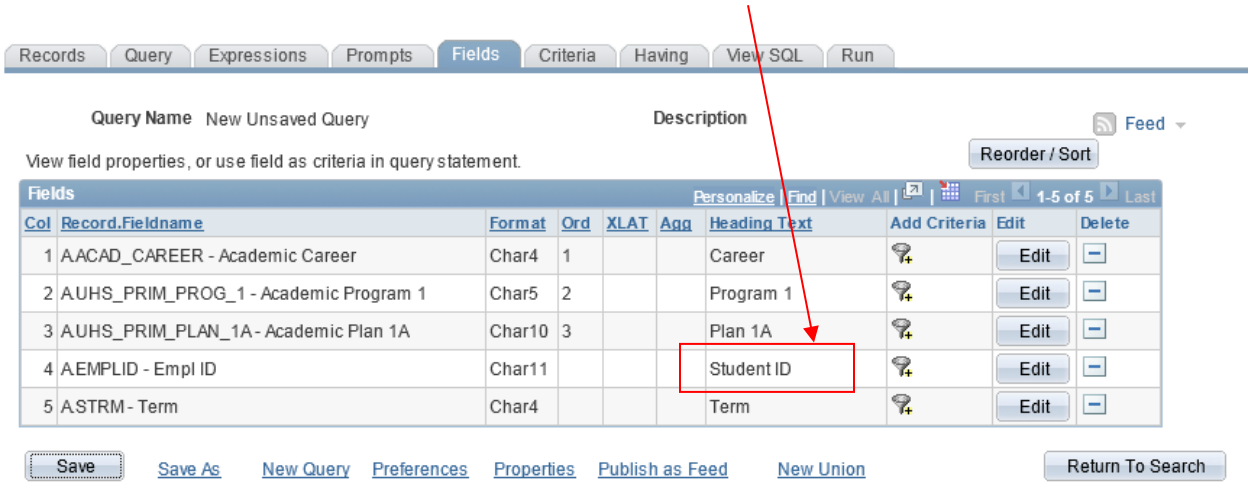


To change the column heading select the heading type or type the desired heading if applicable.

- On the **'Edit Field Properties'** page, select the Heading '**Text**' Radio button and
- Type 'Student ID'.
- Click OK



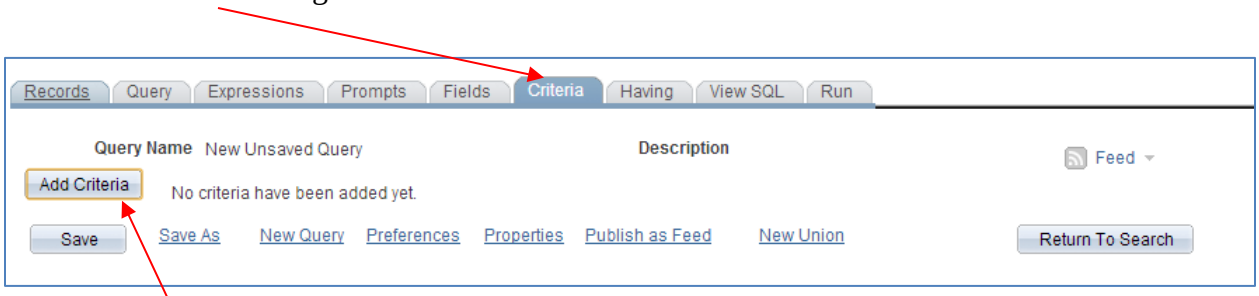
The Fields page will then display the new column heading.

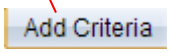


Step 4: Add a Criteria

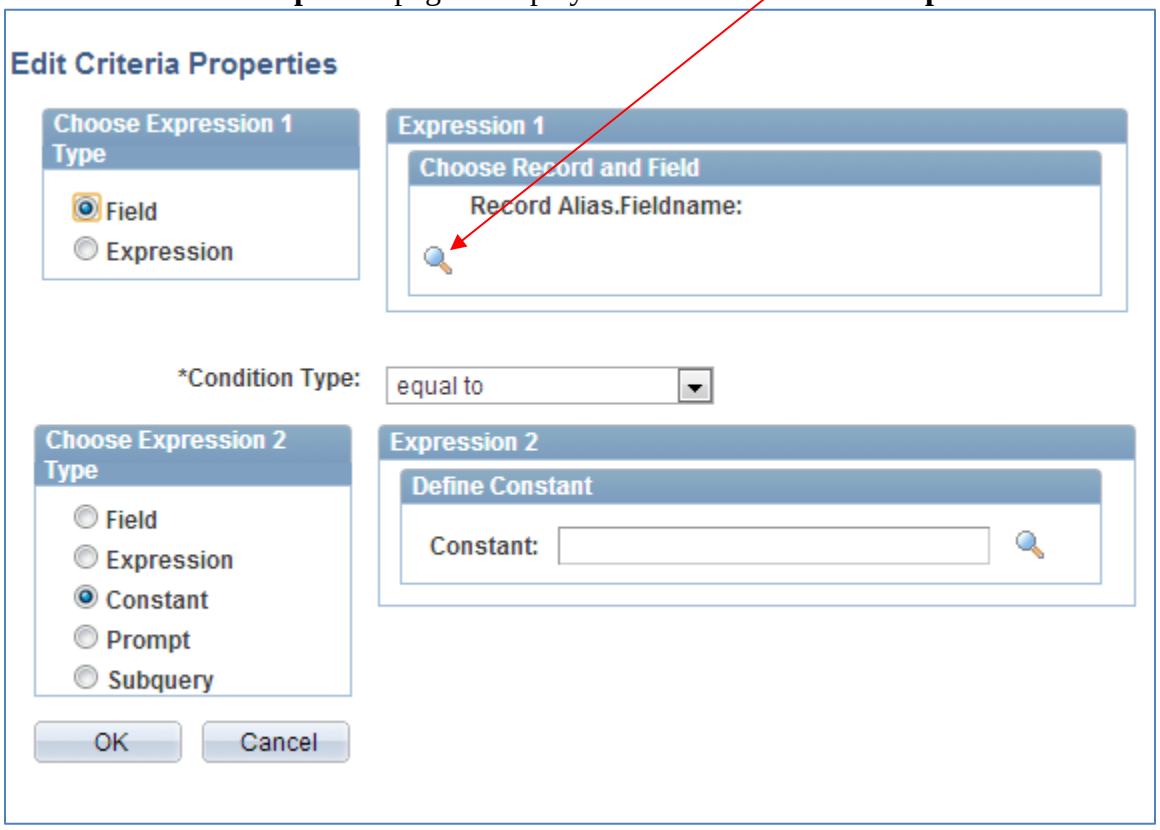
The criteria page is used to specify the specific data to be retrieved by your query. For example, you can limit the students to be returned to a specific institution (i.e. Victoria, Main, Clear Lake).

Click the Criteria tab to go to Criteria .

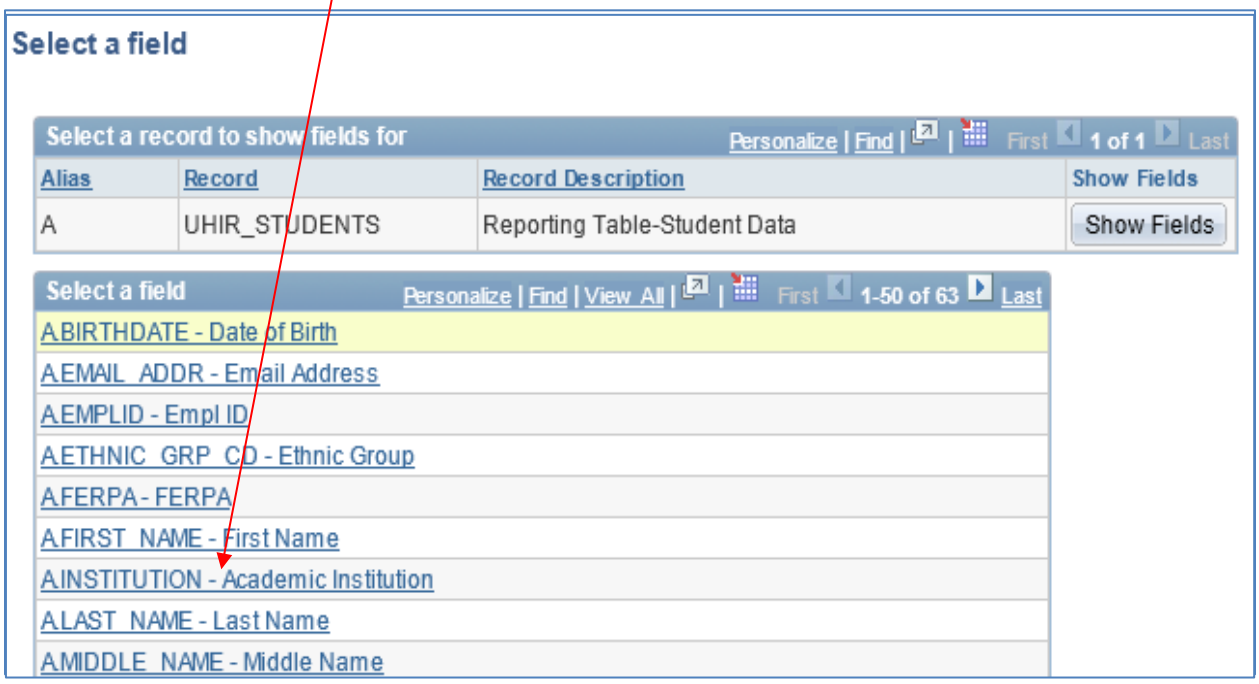


Click on the  button

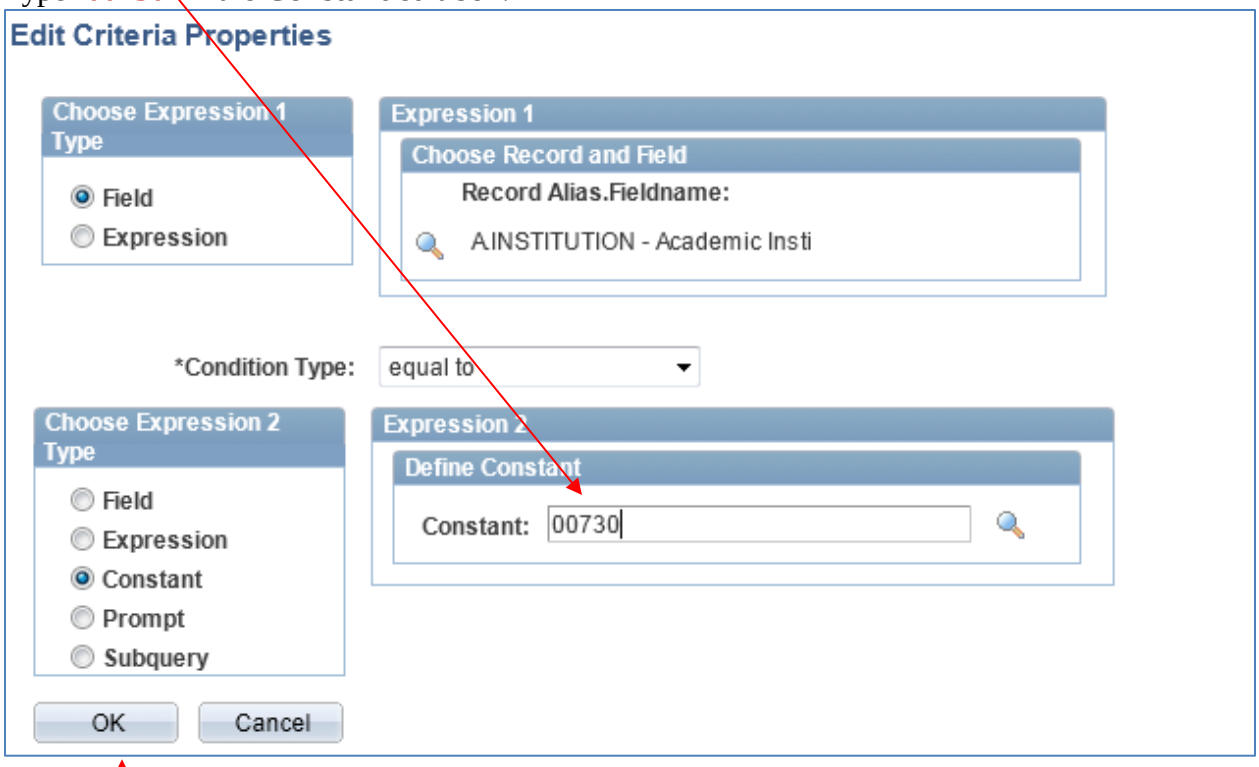
The **Edit Criteria Properties** page is displayed. Click the  in the **Expression 1** area.

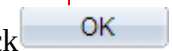


Click on the A.INSTITUTION - Academic Institution link.



Type **'00730'** in the **Constant** edit box.

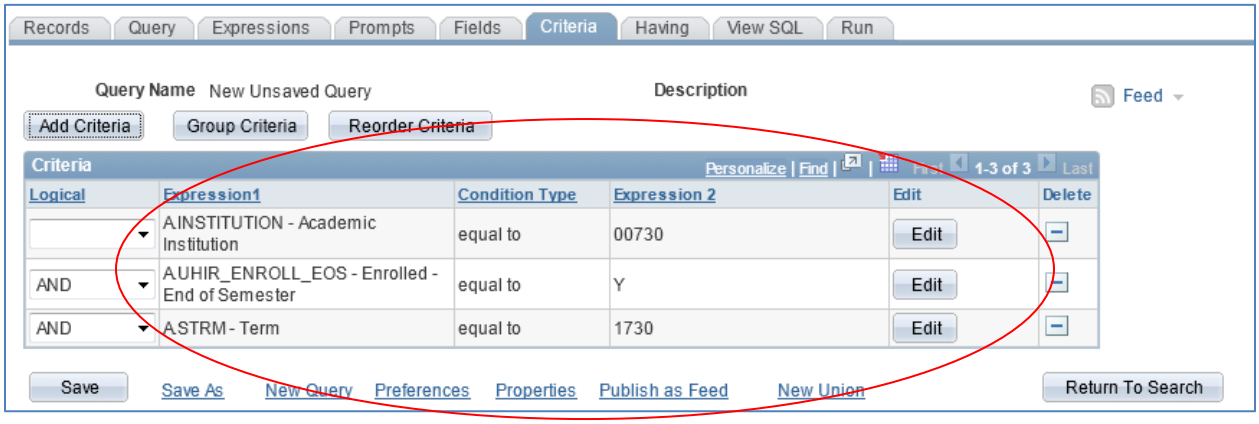


Click 

.

The system returns to the Criteria page and displays the criteria that you previously selected.

Using the same steps above to add criteria, add two more criteria rows to filter out students who were also enrolled on the last day of semester 1730. You should have a total of three criteria rows that look like the below:



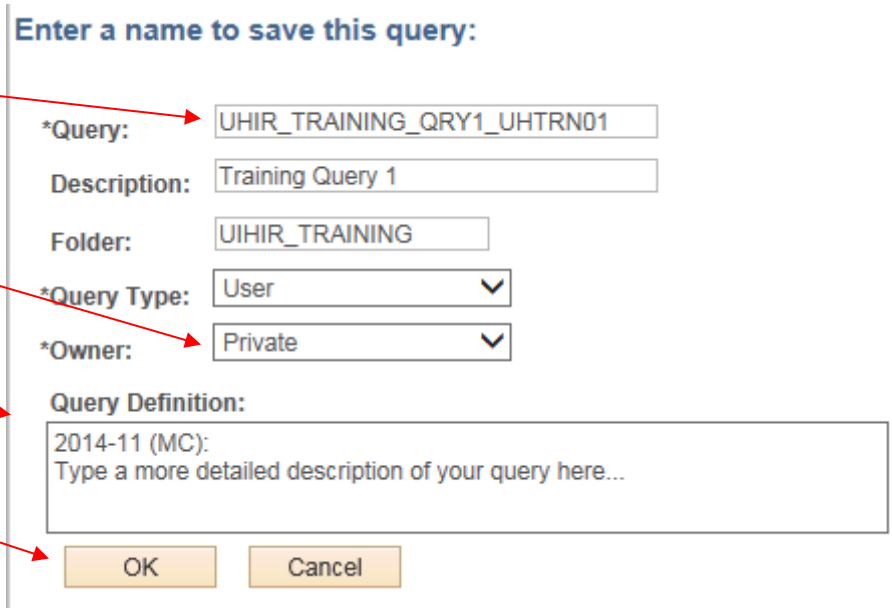
Step 5: Save the Query

(a) Enter the query name **UHIR_TRAINING_QRY1_UHTRN##**

(b) Choose Owner (Private or Public); choose **'Private'**

(c) Enter Query Description and Definition

(d) Click **OK**

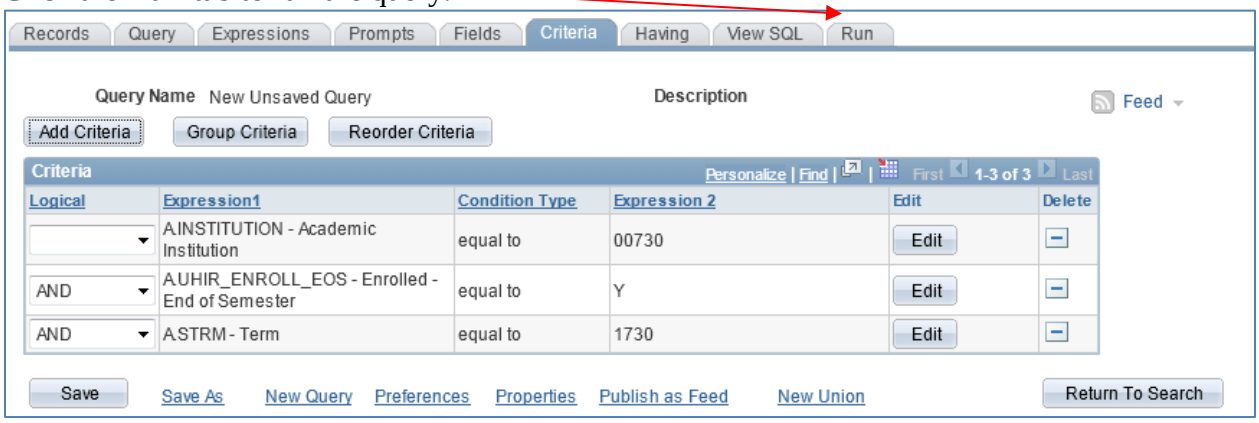


Note: Query names are uppercase and can be up to 30 characters. No spaces or any special characters (other than an underscore) are allowed.

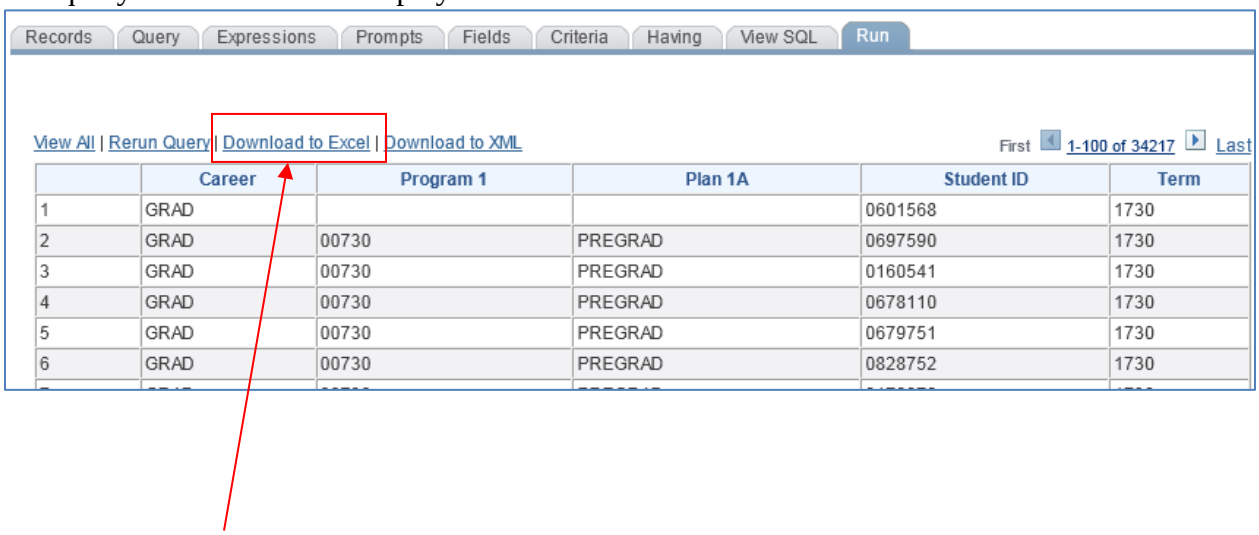
A description for the query can be given, which can be up to 30 characters, mixed case, with special characters

Step 6: Run the Query

Click the **Run** tab to run the query.



The query result set will be displayed.



	Career	Program 1	Plan 1A	Student ID	Term
1	GRAD			0601568	1730
2	GRAD	00730	PREGRAD	0697590	1730
3	GRAD	00730	PREGRAD	0160541	1730
4	GRAD	00730	PREGRAD	0678110	1730
5	GRAD	00730	PREGRAD	0679751	1730
6	GRAD	00730	PREGRAD	0828752	1730

To "Download to Excel", click on the link.

The query results will appear in Microsoft Excel.

	A	B	C	D	E
1		34217			
2	Career	Program 1	Plan 1A	Student ID	Term
3	GRAD			0601568	1730
4	GRAD	00730	PREGRAD	0697590	1730
5	GRAD	00730	PREGRAD	0160541	1730
6	GRAD	00730	PREGRAD	0678110	1730
7	GRAD	00730	PREGRAD	0679751	1730
8	GRAD	00730	PREGRAD	0828752	1730
9	GRAD	00730	PREGRAD	0179079	1730
10	GRAD	00730	PREGRAD	0561523	1730
11	GRAD	00730	PREGRAD	0597414	1730
12	GRAD	00730	PREGRAD	0125938	1730
13	GRAD	00730	PREGRAD	0690299	1730
14	GRAD	00730	PREGRAD	0529923	1730
15	GRAD	00730	PREGRAD	0190205	1730

Practicing copying a private query to another user:

Earlier you learned PeopleSoft Query allows you to perform various actions to a query, one of those is the ability to copy a private query to another user. Copy the query you just created to the UHTRN## user id following your training user id (ie – user UHTRN01 should copy his/her UHIR_TRAINING_QRY1_UHTRN01 query to user id UHTRN02).

In-Class Practice Queries – Chapter 6

Query 1

Create a query that displays the mailing address for all students that were born during 1960 (Hint: Birthdate between 01/01/1960 and 12/31/1960).

Select the following fields: NAME, UHIR_MAIL_ADDRESS1, UHIR_MAIL_ADDRESS2, UHIR_MAIL_CITY, UHIR_MAIL_STATE, UHIR_MAIL_POSTAL, and BIRTHDATE.

Order by NAME. Save the query as UH_MAIL_ADDR_1960.

ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.

Query 2

Create a query that displays the seniors (Hint: **ACAD_LEVEL_BOT = 40**) that were enrolled on the last day of the 1730 semester (Hint: UHIR_ENROLL_EOS = 'Y') at U of H-Main (Hint Institution = '00730') with a Cumulative GPA greater than or equal to 3.0.

Select the following fields in following column order: EMPLID, CUM_GPA, ACAD_CAREER, UHIR_ACAD_LEVEL, UHS_PRIM_PLAN_1A , and INSTITUTION.

Order by CUM_GPA and then by EMPLID. Save the query as UH_SENIORS_3PT.

Query 3

Create a list of students that have received a degree at the University of Houston – Main (Hint: UHS_DEGREE_INST = '00730') in 2006 or 2007.

Select the following fields: EMPLID, STDNT_DEGR, UHIR_DEGR_CAREER, UHIR_DEGR_DESCR, UHIR_DEGR_DT, UHIR_DEGR_MAJOR1

Order by UHIR_DEGR_DT and then by UHIR_DEGR_CAREER. Save the query as UH_DEGREE_20062007.

ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.

VOLUME III:

Advanced PeopleSoft Query Writing

Objectives

Participants will learn the following concepts and procedures:

- Understand how to incorporate advanced query features
    - Aggregate Functions
    - Having Criteria
    - Prompts
    - Expressions
    - Join Multiple Tables
    - Subqueries
    - Unions

Overview

Advanced query writing functionality available in the PeopleSoft Query tool will be incorporated into query writing. Users will build upon basic queries used in previous training by adding more complex features such as aggregation, expression functions, joining multiple tables, and more.

CHAPTER 7 – Creating Queries with Aggregate

Overview

In this chapter, you will learn how to utilize a query that uses an aggregate function.

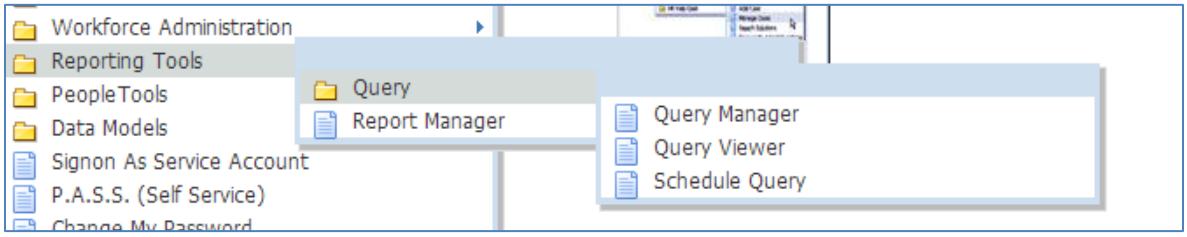
In a standard query, each row in the result set corresponds to an individual row in the table that you are querying. Sometimes, however, you instead want a summary of the information in multiple rows. For example, you might want to know how many students are currently associated to the programs. You can query for this kind of summary information using aggregate functions.

An aggregate function is a special type of operator that returns a single value based on multiple rows of data. When your query includes one or more aggregate functions, PeopleSoft Query collects related rows and displays a single row that summarizes their contents.

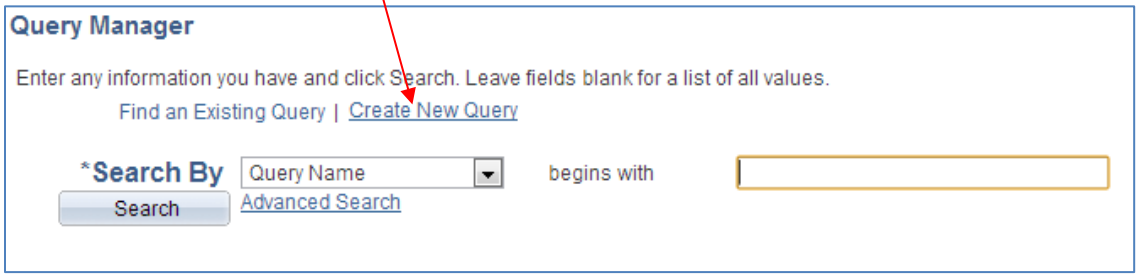
Example

List the number of students within each academic career, display the career as the first column of output.

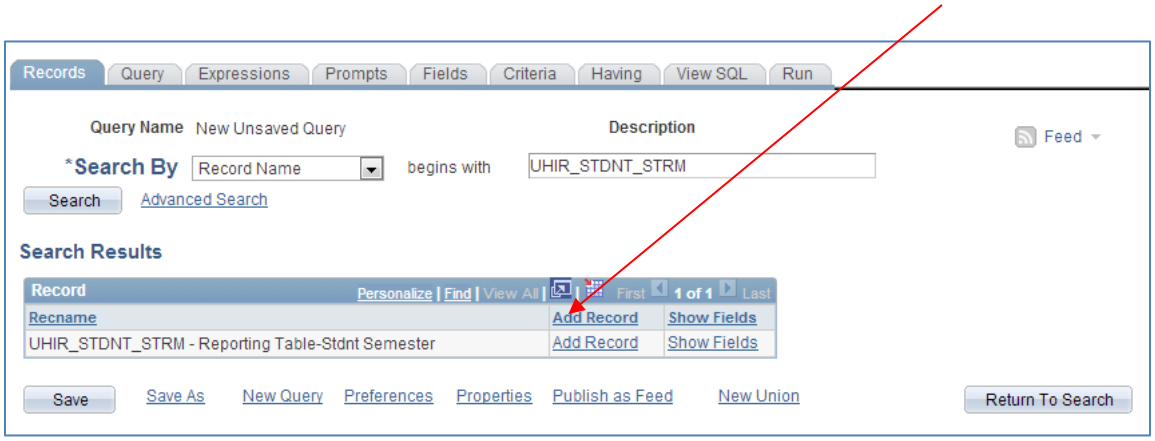
Step 1: Navigate to Query Manager



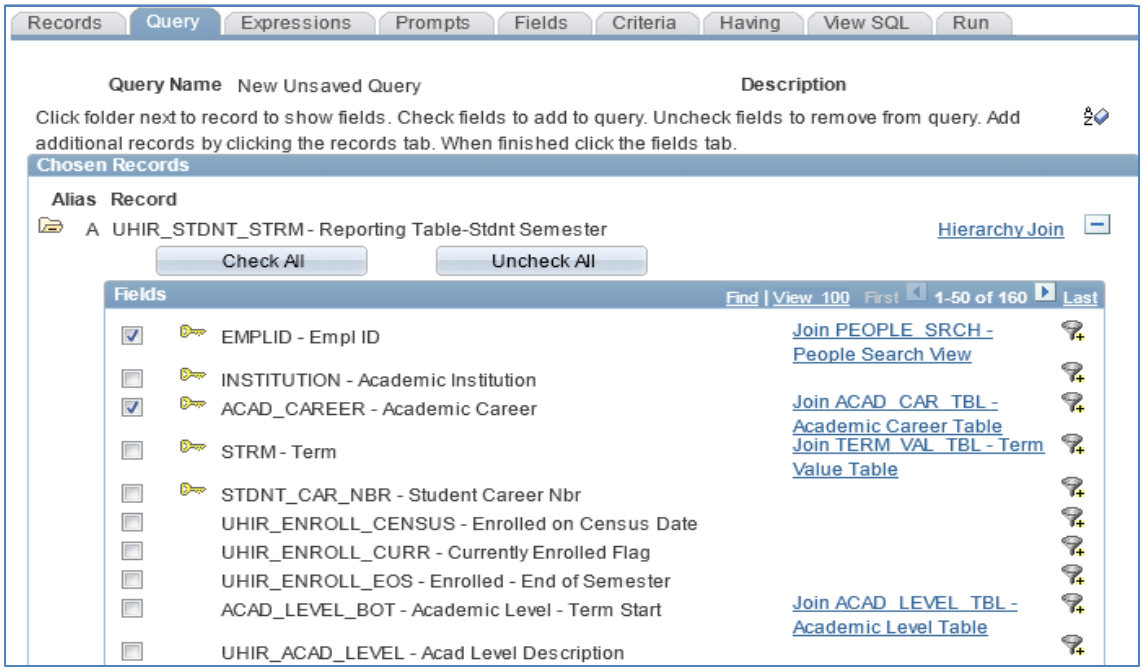
Step 2: Click on Create New Query **.**



Step 3: Add the UHIR_STDNT_STRM record and Click "Add Record".

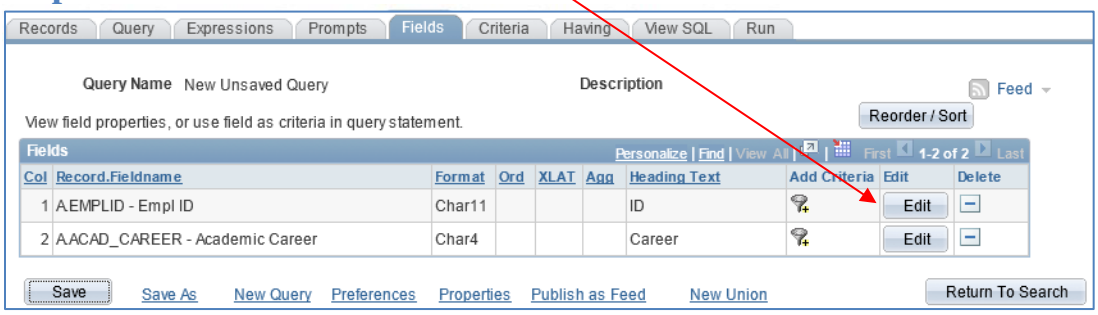


Step 4: Select the EMPLID and ACAD_CAREER fields

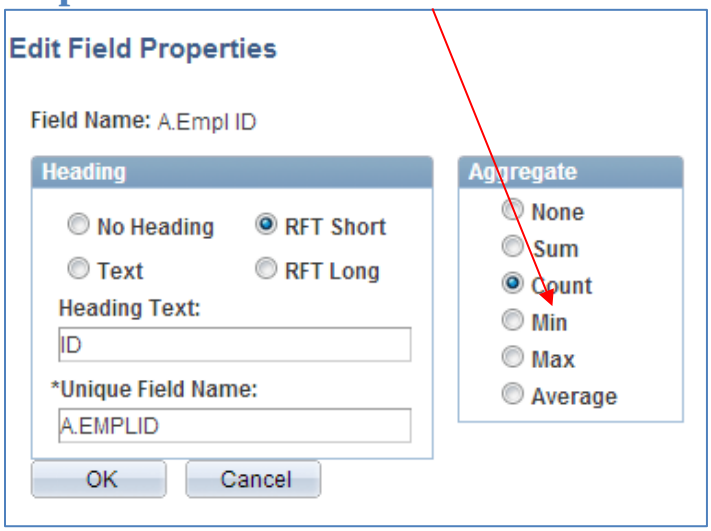


Step 5: Click on the 'Fields' Tab to view the Fields Page.

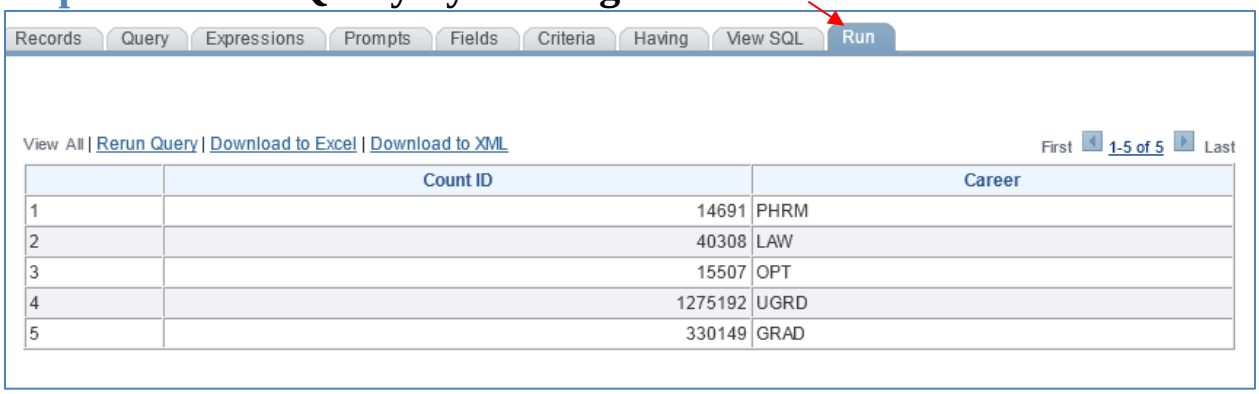
Step 6: Click on the EMPLID Edit button.



Step 7: Click on the 'Count' Radio Button and Click OK.



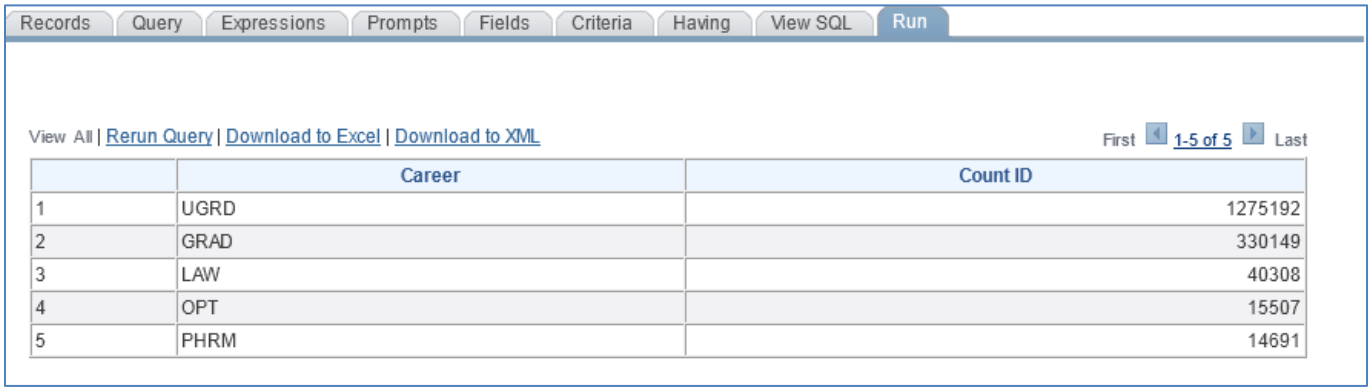
Step 8: Run the Query by clicking on the 'Run' tab.



	Count ID	Career
1	14691	PHRM
2	40308	LAW
3	15507	OPT
4	1275192	UGRD
5	330149	GRAD

Step 9: Resort the data to display the Career with the Student Counts in descending order.

Step 10: Rerun the query and view the results. ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION. Save the query as UH_CAREER_STDNT_CNT.



	Career	Count ID
1	UGRD	1275192
2	GRAD	330149
3	LAW	40308
4	OPT	15507
5	PHRM	14691

Chapter 7: In-Class Exercise (Aggregate Functions)

Query 1

Create a query that returns a list of primary plans (UHS_PRIM_PLAN_1A) and the number of students in each of those plans for students that enrolled at UH-Main during the Fall-2007 semester.

Sort the plans from the plan with the most to least number of students in the plan. The primary plans should be displayed as the first column of output.

Save the query using the below name and ownership.

Query Name: **UH_PLAN_STDNT_CNT**
Owner: **Private**

ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.

Sample Output

View All | Rerun Query | Download to Excel | Download to XML First 1-100 of 540 Last

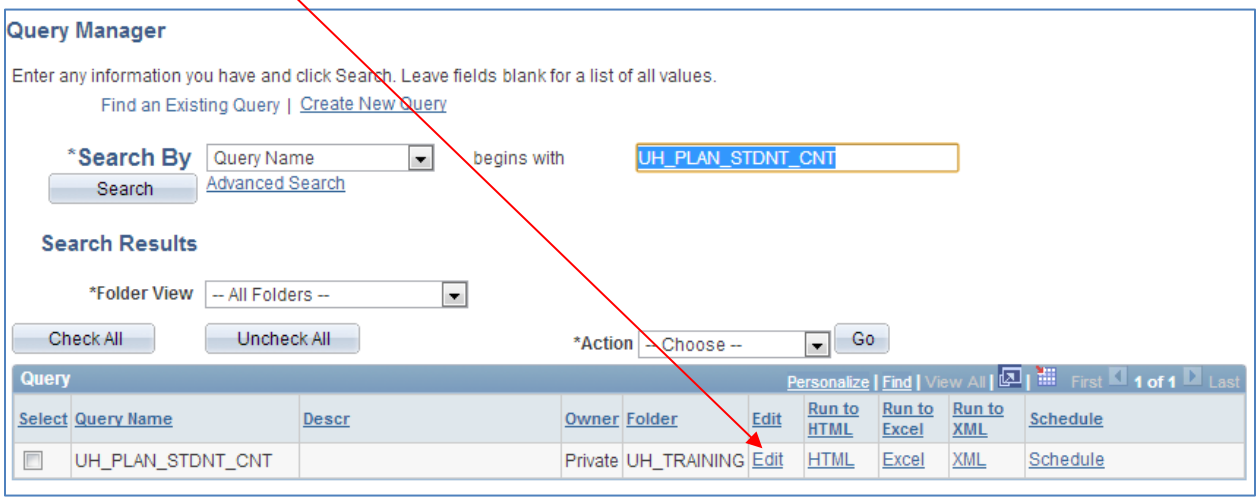
	Plan 1A	Count ID
1	USDDEU	4767
2	PREBDEU	2157
3	BIOBS	1320
4	LAWJD	1004
5	INDISTBS	999
6	HRMGTBS	927
7	BADMMBA	813
8	ACCBBA	729
9	PREPDEU	717
10	ARCBARCH	627
11	FINBBA	584
12	PRPSDEU	581
13	PSYCBA	511
14	PHARMPHRD	511
15	HISTBA	475
16	MEBSME	470
17	POLSBA	412
18	BIOBA	410
19	OPTOD	407
20	PSYCBS	405

CHAPTER 8 – Creating Query with Having Criteria

Overview In this chapter, you will learn how to add a having clause to limit the number of records on the report based on a specific count.

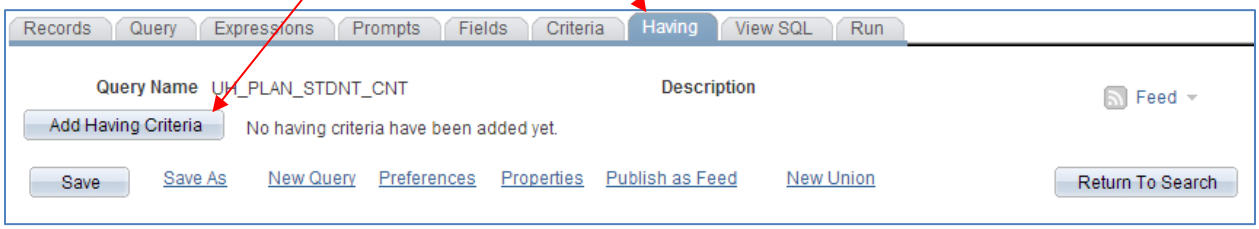
Example *The UH_PLAN_STDNT_CNT query will be restricted to only display plans that have more than 100 students.*

Step 1: Click Edit to modify the query.

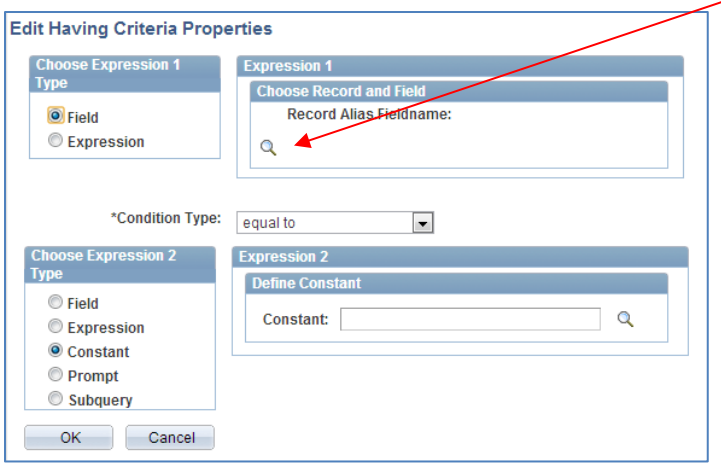


Step 2: Click on the 'Having' Page

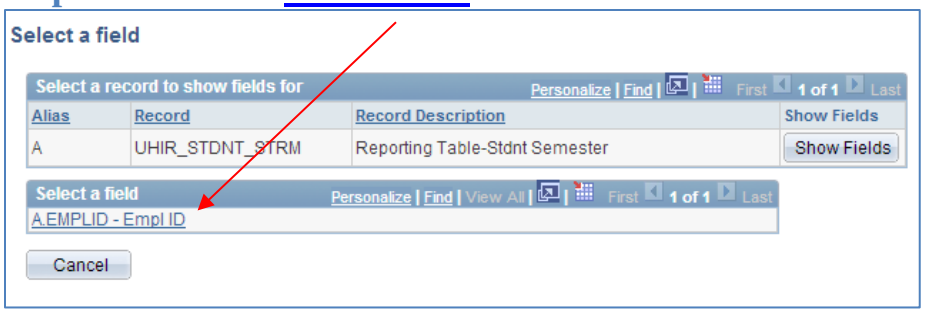
Step 3: Click on the Add Having Criteria **button.**



Step 4: Click on the magnifying glass icon for Expression 1

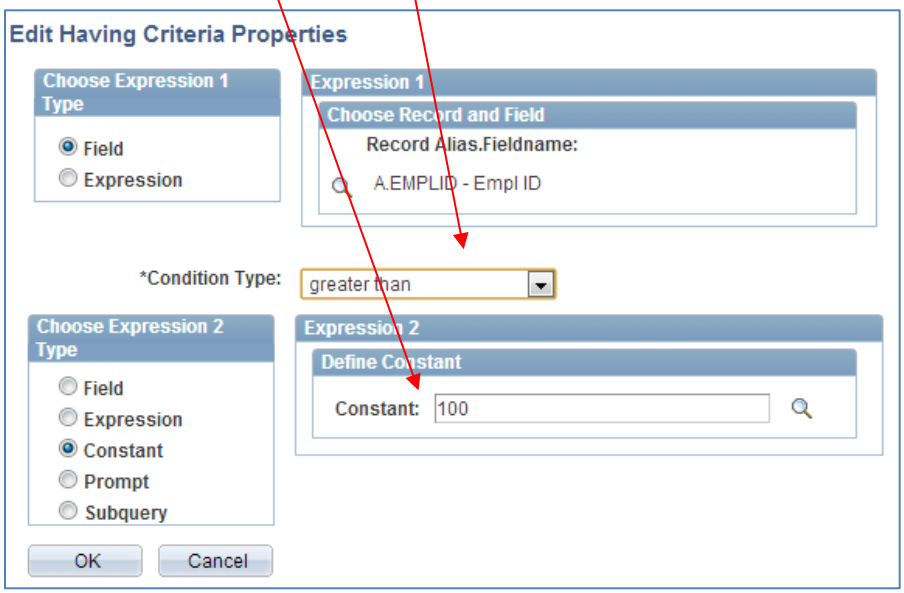


Step 5: Select the A.EMPLID link.

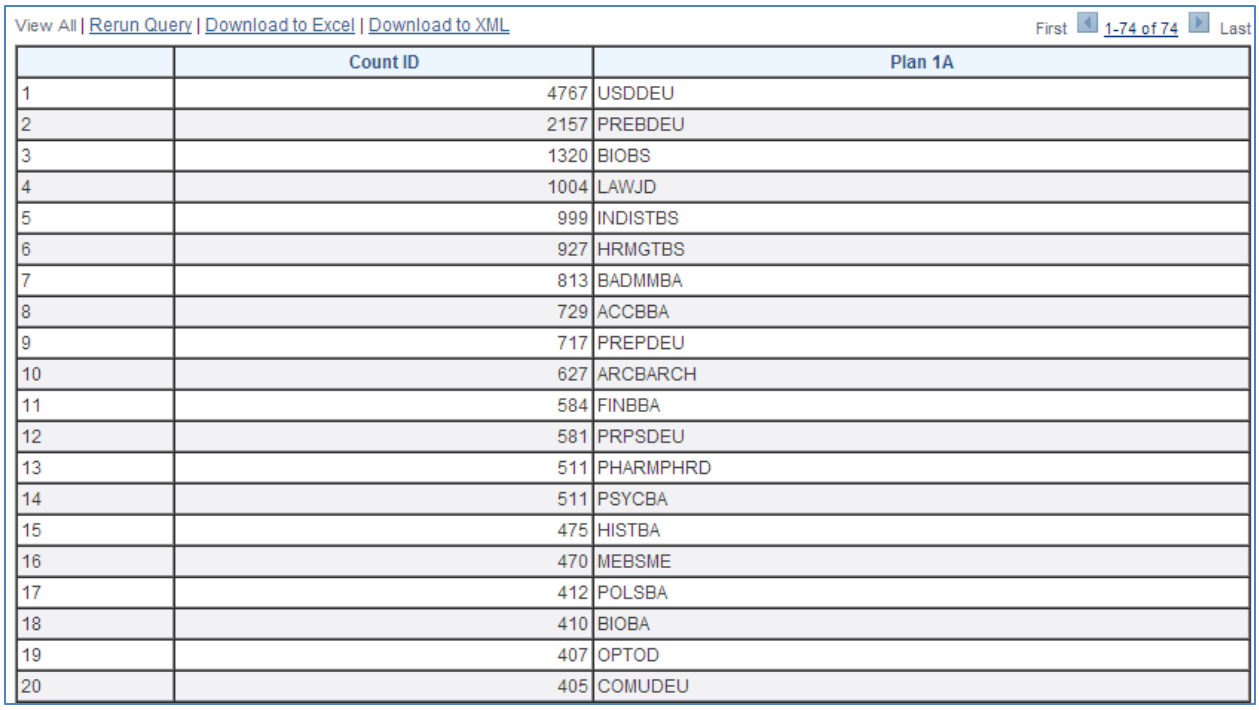


Step 6: Select 'greater than' Condition Type.

Step 7: Enter '100' in the Expression 2 Constant Edit Box and click OK.



Step 8: Run the Query by clicking on the 'Run' tab.



View All | Rerun Query | Download to Excel | Download to XML

First 1-74 of 74 Last

	Count ID	Plan 1A
1	4767	USDDEU
2	2157	PREBDEU
3	1320	BIOBS
4	1004	LAWJD
5	999	INDISTBS
6	927	HRMGTBS
7	813	BADMMBA
8	729	ACCBBA
9	717	PREPDEU
10	627	ARCBARCH
11	584	FINBBA
12	581	PRPSDEU
13	511	PHARMPHRD
14	511	PSYCBA
15	475	HISTBA
16	470	MEBSME
17	412	POLSBA
18	410	BIOBA
19	407	OPTOD
20	405	COMUDEU

Note: Only plans with more than 100 students associated to them are displayed because the **having clause** restricts the plans listed to those that have over 100 students.

Step 9: ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION. Save the query as UH_PLAN_STDNT_CNT2.

Chapter 8: In-Class Exercise (Having Criteria)

Query 1

Create a query that displays primary programs (UHS_PRIM_PROG_1) having at least 1,000 students in the program who were enrolled at UH-Main during the Fall 2008 semester. Order the programs by count of students in descending order. Save the query as UH_PROG_STDNT_CNT.

ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.

CHAPTER 9 – Creating Queries with Prompts

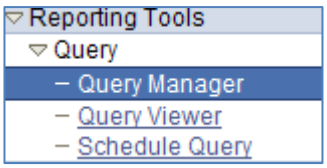
Overview

In this chapter, you will learn how to set up run-time prompts, in which a User enters a value for a specific field at the time he executes the query. You can set up run-time prompts as criteria in your queries.

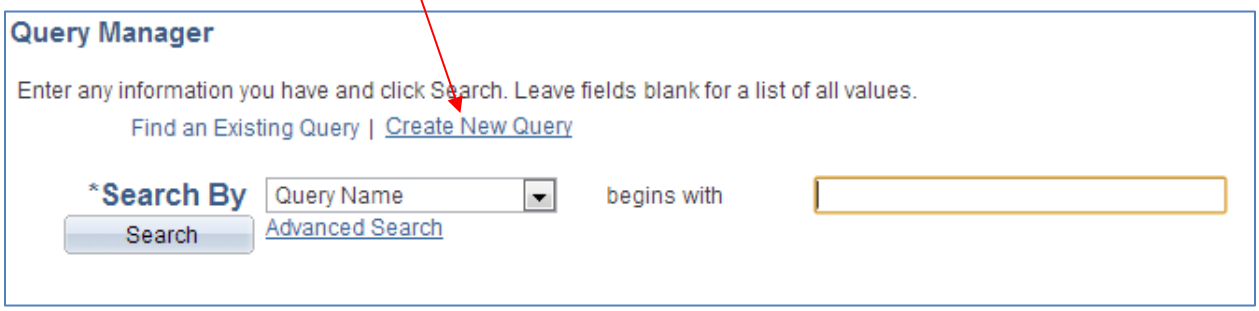
Example

Create a query that lists the UHM students that were in a specific Academic Program in the fall of 2007 as of census. The academic program should be a run time prompt so that the user running the query can select the program at run-time.

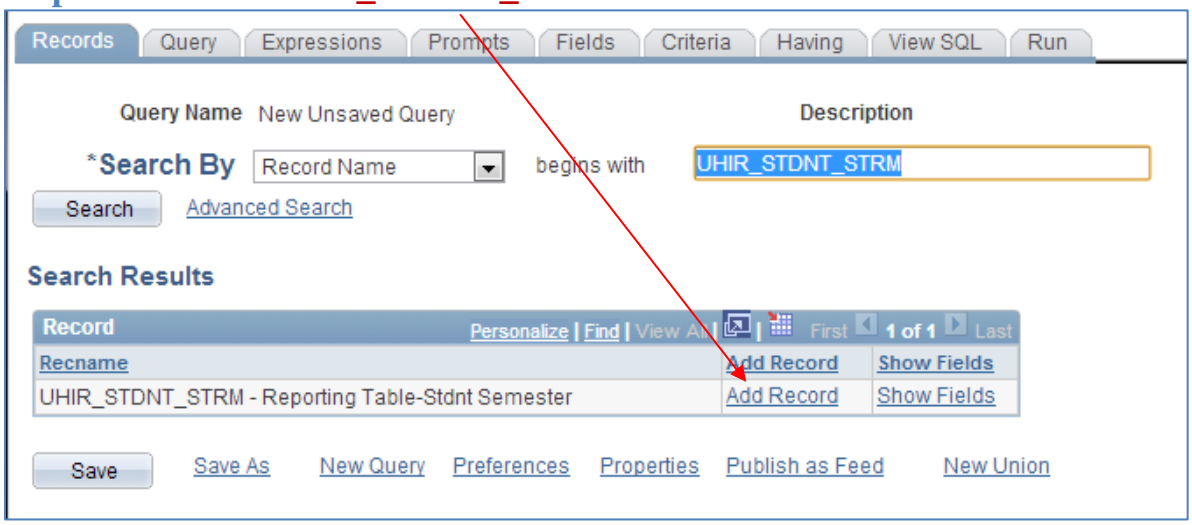
Step 1: Navigate to Query Manager



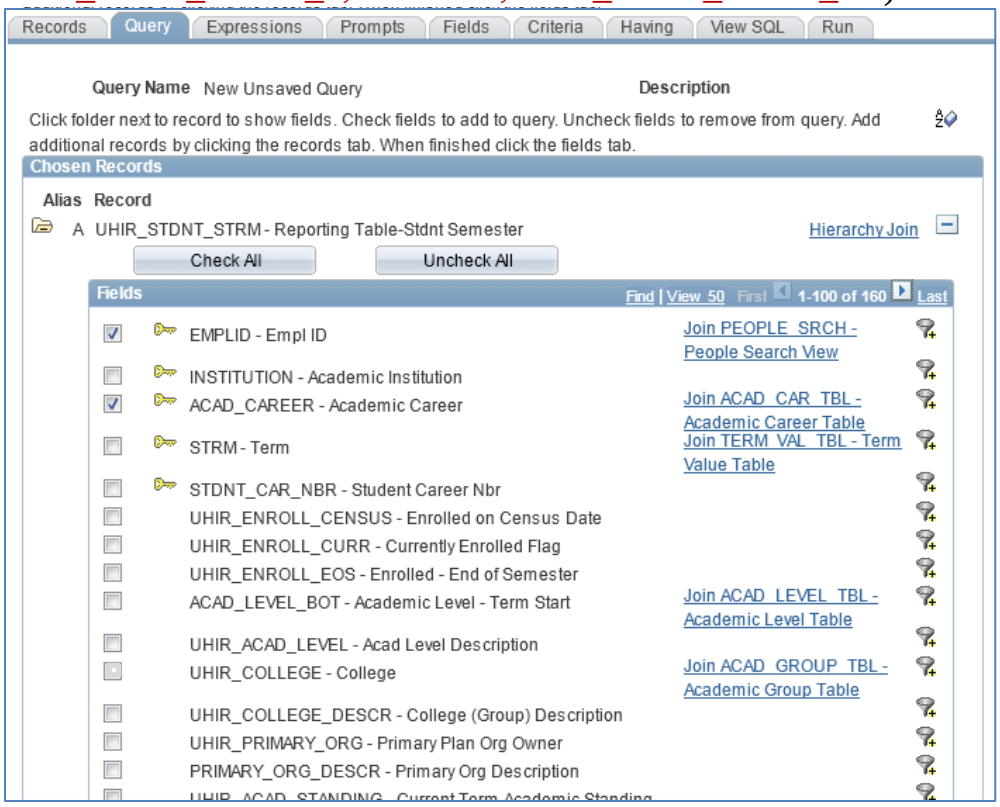
Step 2: Click on Create New Query **.**



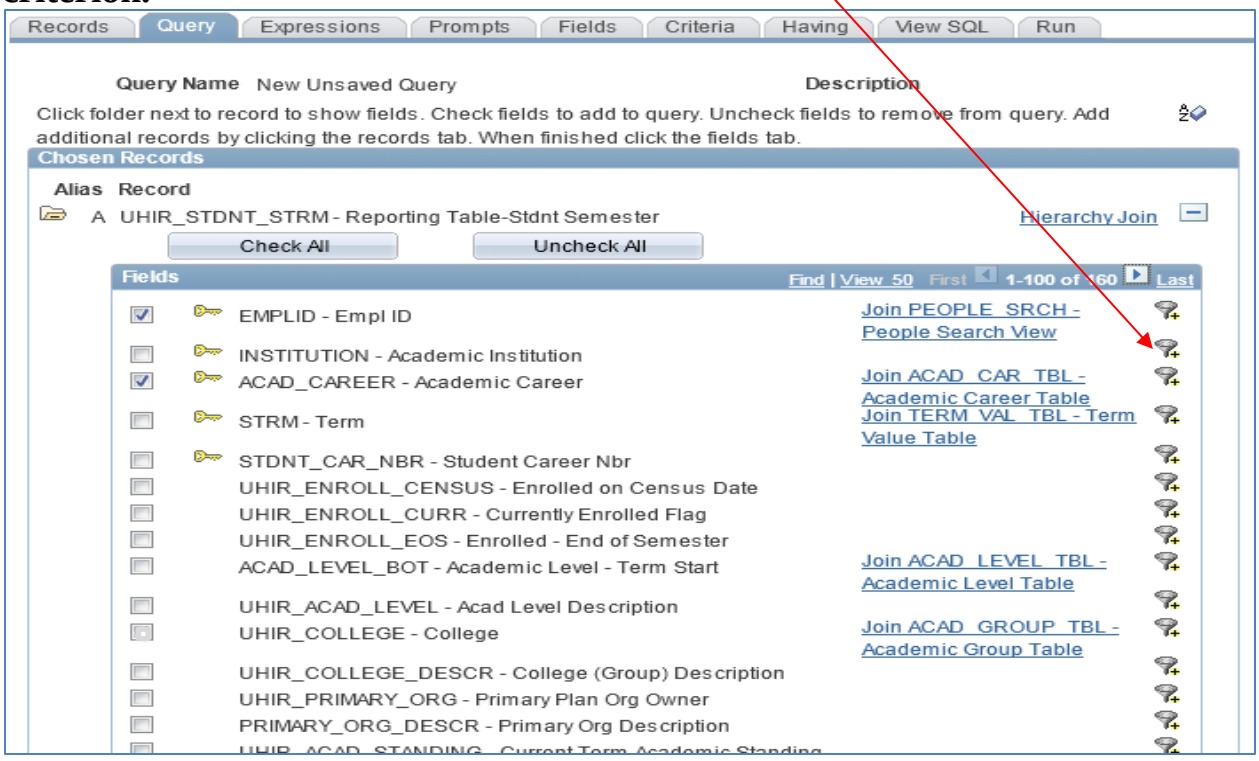
Step 3: Add the UHIR_STDNT_STRM record.



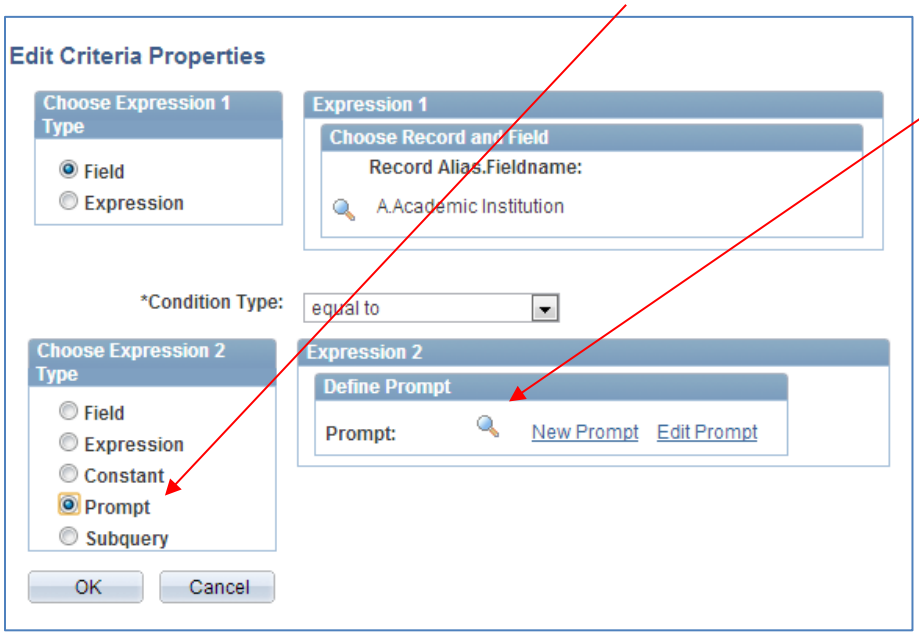
Step 4: Select the fields that you want listed on the report (ACAD_CAREER, UHS_PRIM_PROG_1, EMPLID, UHS_PRIM_PLAN_1A). ***Use also for Sort Order.***



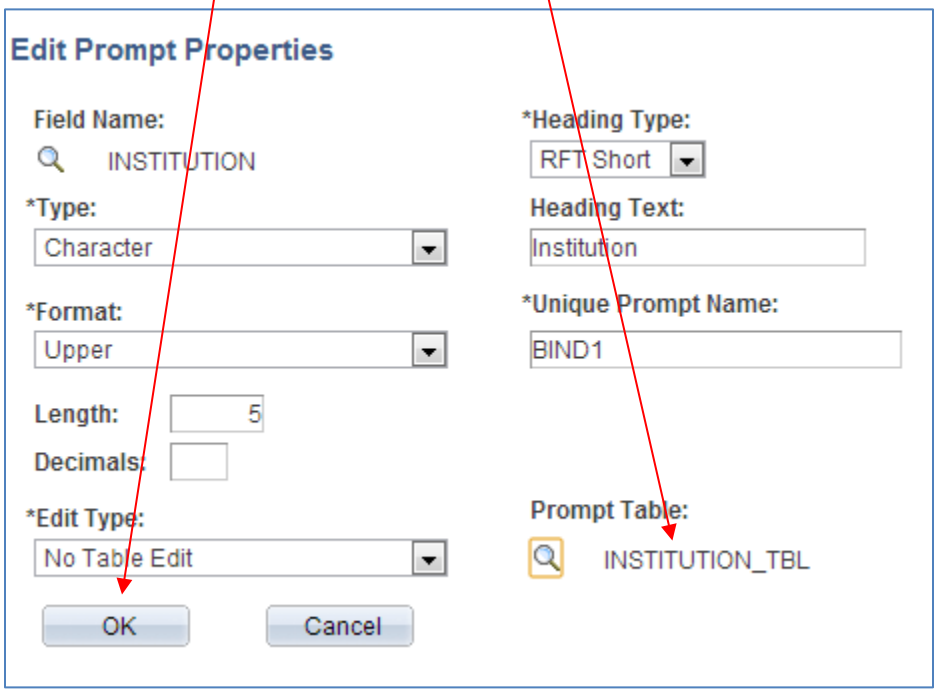
Step 5: Click on the  **next to the ACADEMIC INSTITUTION field to create a criterion.**



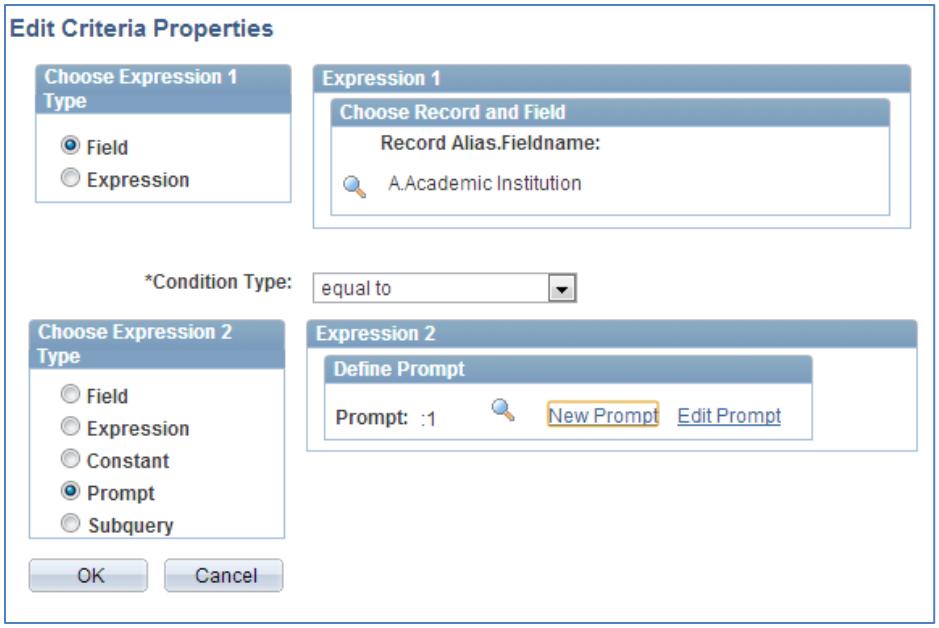
Step 6: Click the Expression 2 Type of 'Prompt' and click on New Prompt.



Step 7: Select the INSTITUTION_TBL as the Prompt Table if not already selected and click OK **.**

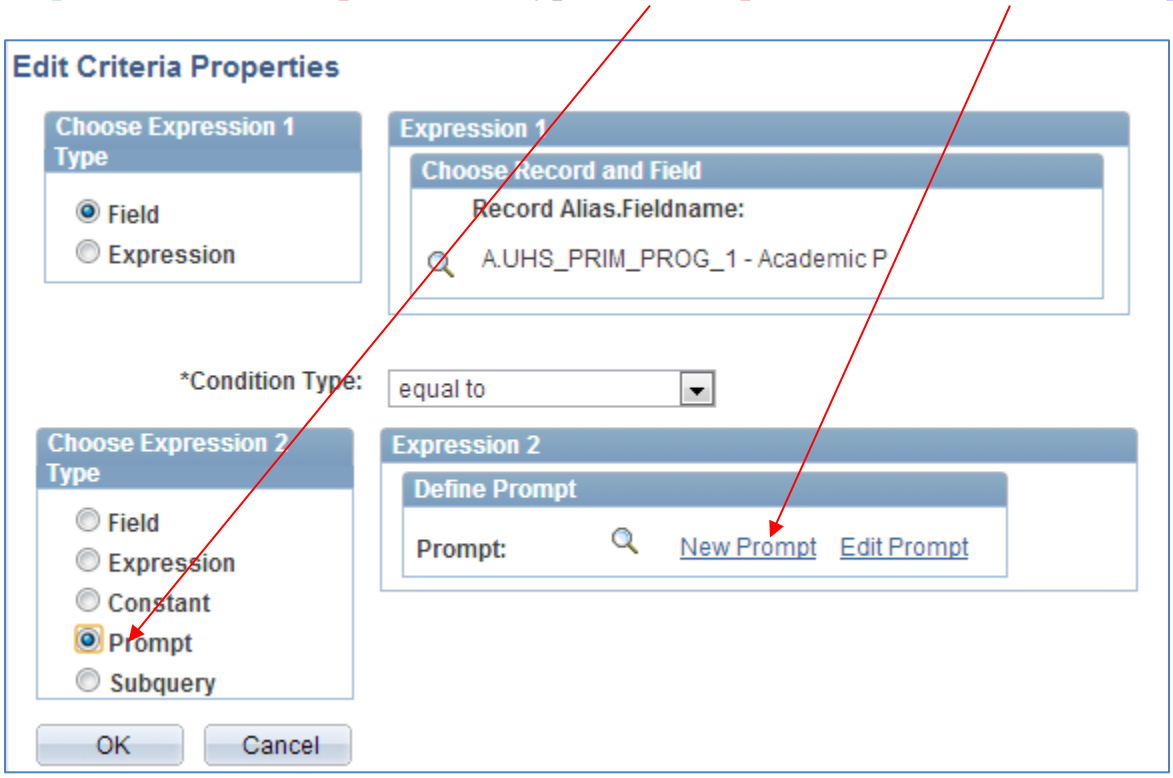


Step 8: Click OK **on the Edit Criteria Properties page.**

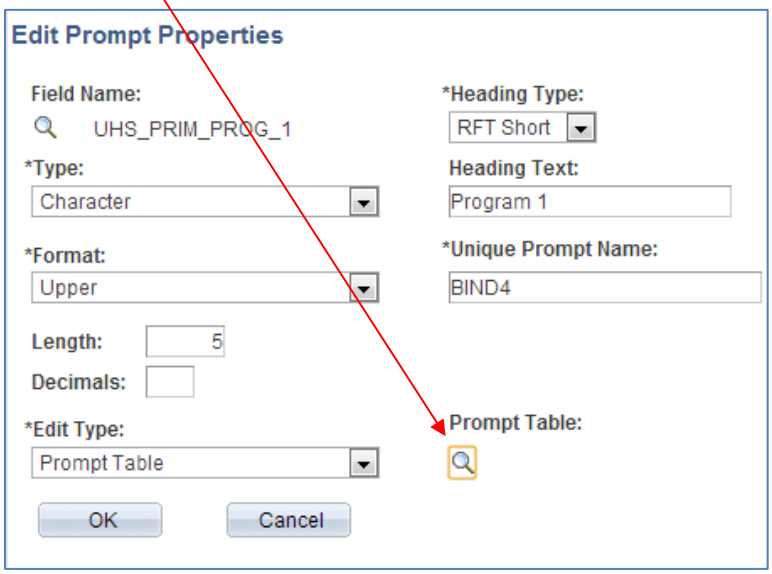


Step 9: Select Add Criteria next to the UHS_PRIM_PROG_1 field.

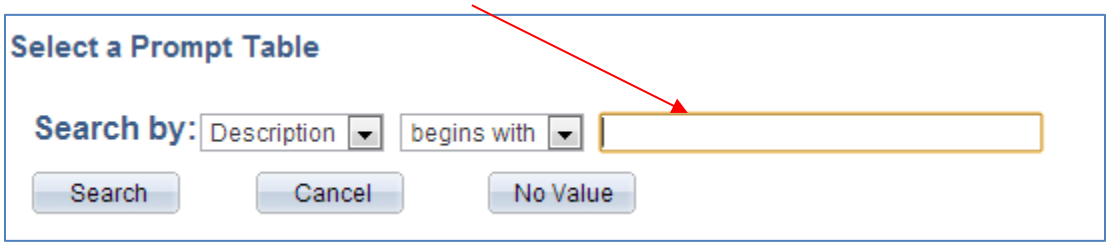
Step 10: Click the Expression 2 Type of 'Prompt' and click on New Prompt.



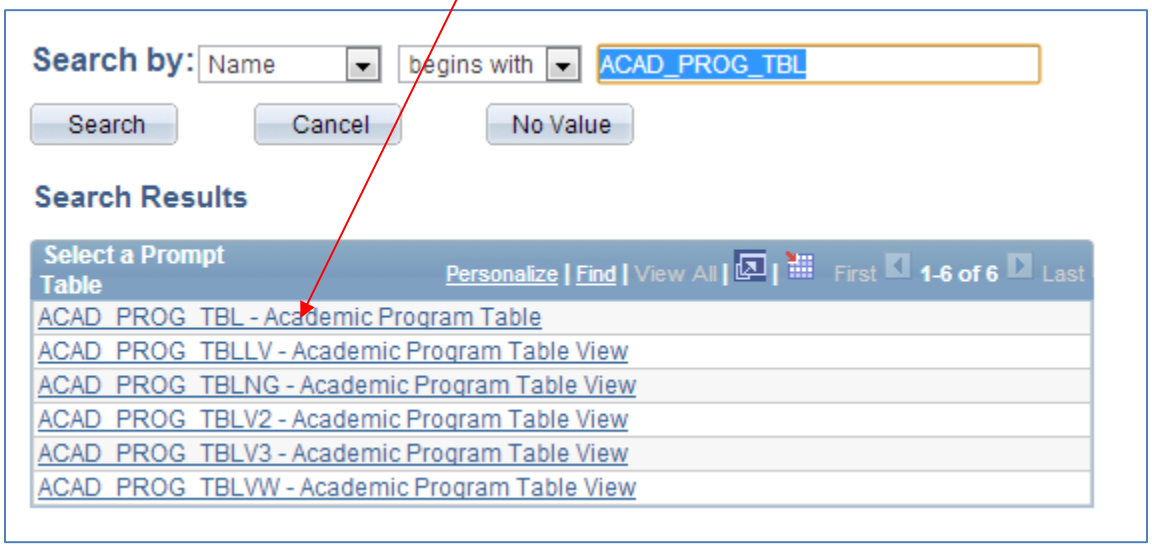
Step 11: Select the magnifying glass below the Prompt Table.



Step 12: Enter value 'ACAD_PROG_TBL' and click Search.

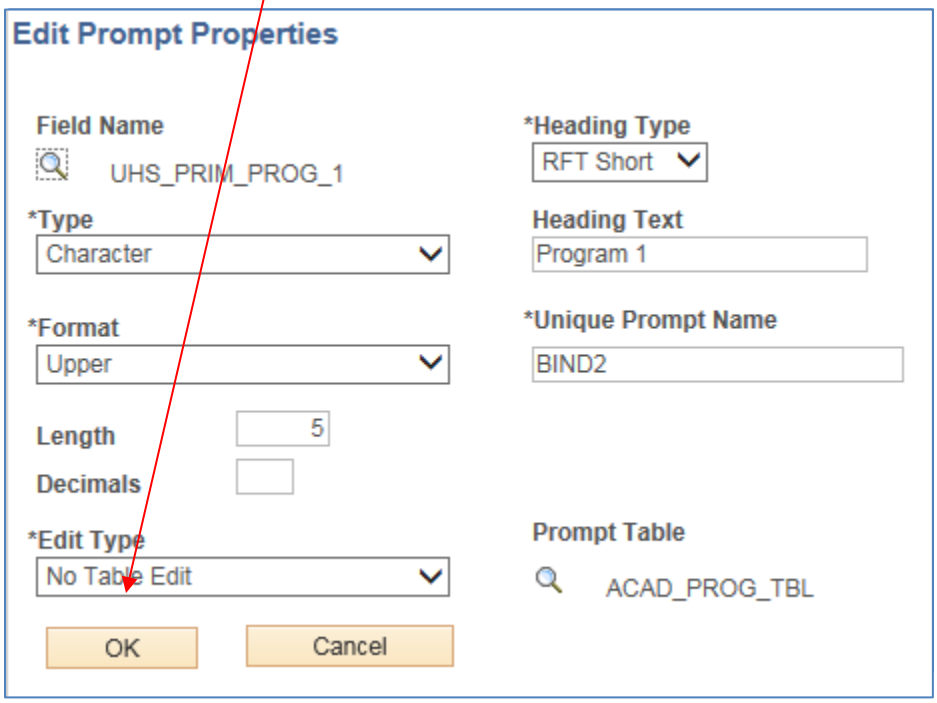


Step 13: Click on the ACAD_PROG_TBL link.



Step 14: Enter '**030UG**' in the Default Value field.

Step 15: Click OK on the **Edit Prompt Properties** page.



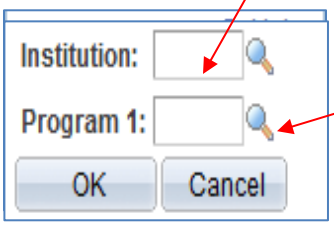
Step 16: Add a constant criteria so that **STRM = '1730'**.

Step 17: Add a constant criteria so that **UHIR_ENROLL_CENSUS = 'Y'**.

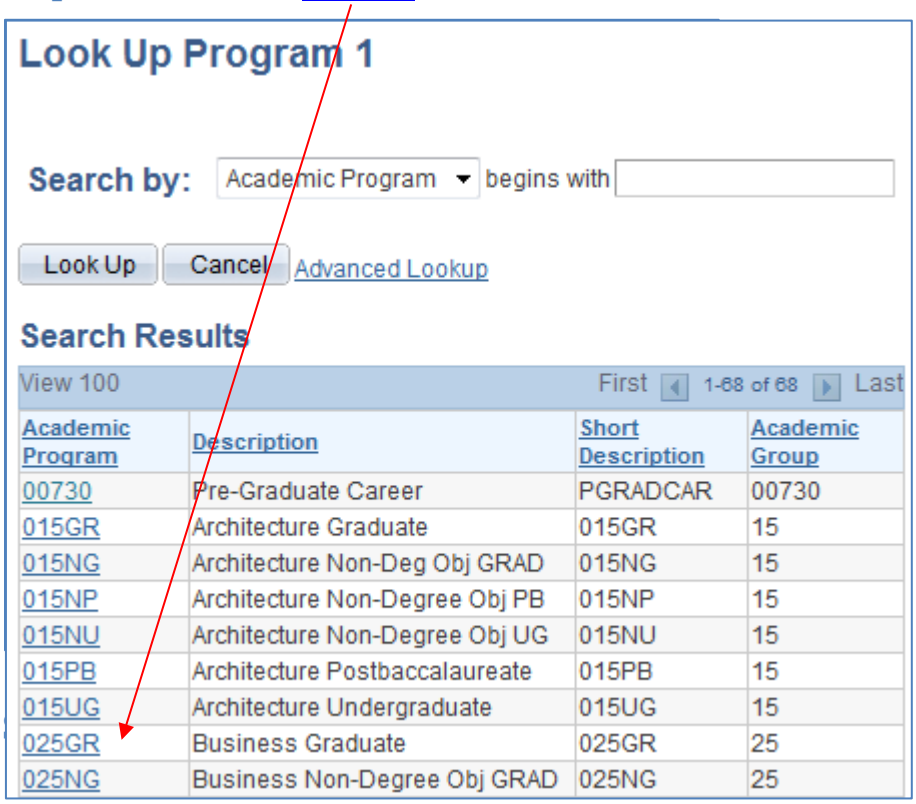
Step 18: Run the Report by clicking on the '**Run**' Tab.

Step 19: Enter '**00730**' in the Institution Field.

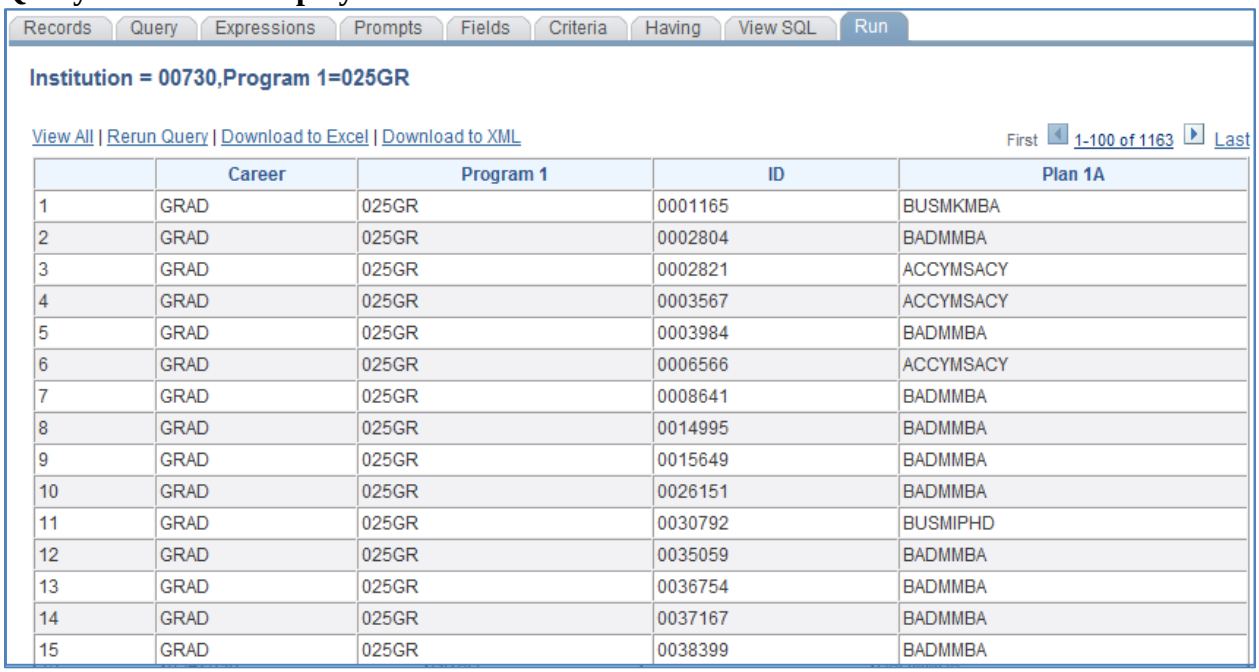
Step 20: Note the defaulted value; click on the magnifying glass next to **Program1**.



Step 21: Select the 025GR Link.

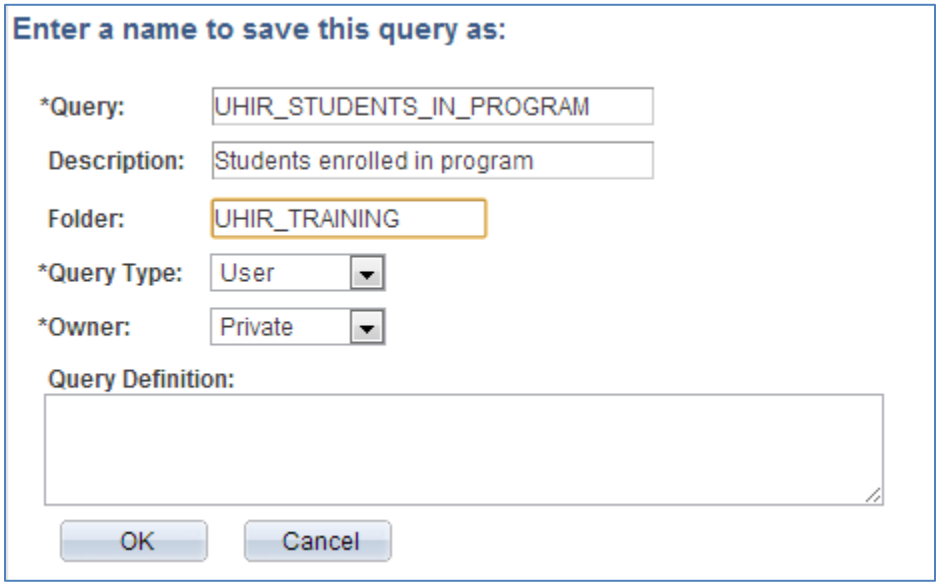


Query Results are displayed



	Career	Program 1	ID	Plan 1A
1	GRAD	025GR	0001165	BUSMKMBA
2	GRAD	025GR	0002804	BADMMBA
3	GRAD	025GR	0002821	ACCYMSACY
4	GRAD	025GR	0003567	ACCYMSACY
5	GRAD	025GR	0003984	BADMMBA
6	GRAD	025GR	0006566	ACCYMSACY
7	GRAD	025GR	0008641	BADMMBA
8	GRAD	025GR	0014995	BADMMBA
9	GRAD	025GR	0015649	BADMMBA
10	GRAD	025GR	0026151	BADMMBA
11	GRAD	025GR	0030792	BUSMIPHD
12	GRAD	025GR	0035059	BADMMBA
13	GRAD	025GR	0036754	BADMMBA
14	GRAD	025GR	0037167	BADMMBA
15	GRAD	025GR	0038399	BADMMBA

Step 22: Save the query as 'UHIR_STUDENTS_IN_PROGRAM'.



Step 23: (Observation Exercise for "Optional" prompting) Rerun the query and leave the run-time variable institution Blank, you should get zero rows of data returned. Go back and change the Institution run-time prompt by selecting the "Optional" checkbox; rerun the query with leaving institution Blank again, this time you should now get back rows of data.

Chapter 9: In-Class Exercise (Adding Prompts)

Query 1

Instead of creating a new query from scratch, use an existing query as the basis of your query definition since it has most of what's needed for this query exercise. Remove all criteria from the existing query. Add the following prompts to the UH_PLAN_STDNT_CNT2 query:

Prompt 1: INSTITUTION (Prompt Table INSTITUTN_SCRTY) (set default to 00730)
Prompt 2: STRM (Prompt Table TERM_VAL_TBL)

Run the query for UH-Main, term 1740. Instead of overwriting the previous query definition, perform a Save As command and name the query UH_PLAN_STDNT_CNT3. **ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.**

Query 2

Using existing query UH_PLAN_STDNT_CNT as the starting point for your extract, add a variable that prompts for the Academic Career at run-time. Run the query for the undergraduate career 'UGRD'…Note you get an error trying to run the query.

What should you do to be able to use the prompt on the academic career field?...Make the necessary adjustments with the instructor and rerun the query. Instead of overwriting the previous query definition, perform a Save As command and name the query UH_PLAN_STDNT_CNT4. **ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.**

Query 3

List the number of students per Primary Academic Program per Academic Career. The user will need to enter the desired careers and term for the output at the time the query is run.

Display in order the Academic Career, Primary Academic Program, Primary Academic Program Description, and number of Students. The data should be sorted for the user by Academic Career in ascending order and then by number of students in descending order. Two variables are needed for careers and a single variable is needed for the term. The first run-time variable for career should use custom text "Enter First Career" and the second run-time variable for career should use custom text "Enter Second Career".

Run the query for UH-Main, careers Law (LAW) and Optometry (OPT), and term Fall 2008. Name the query UH_CAREER_PROG_CNT. **ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.**

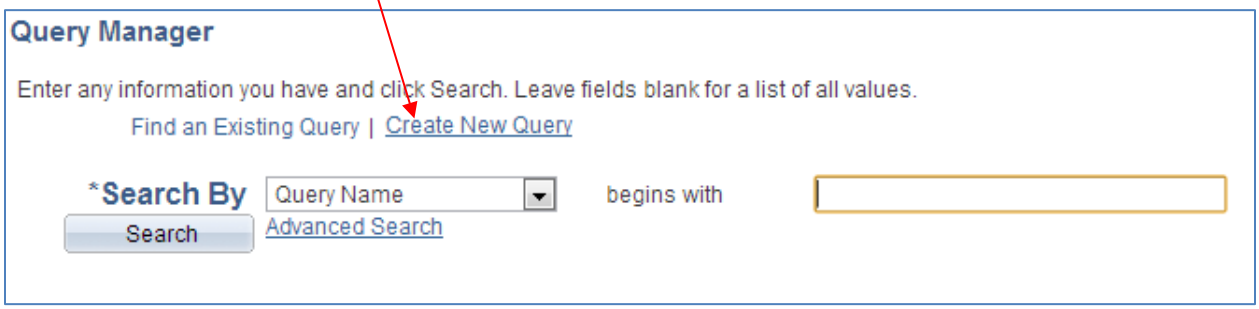
CHAPTER 10 – Creating Expressions

Overview In this chapter, you will learn how to create an expression to combine multiple fields into one field.

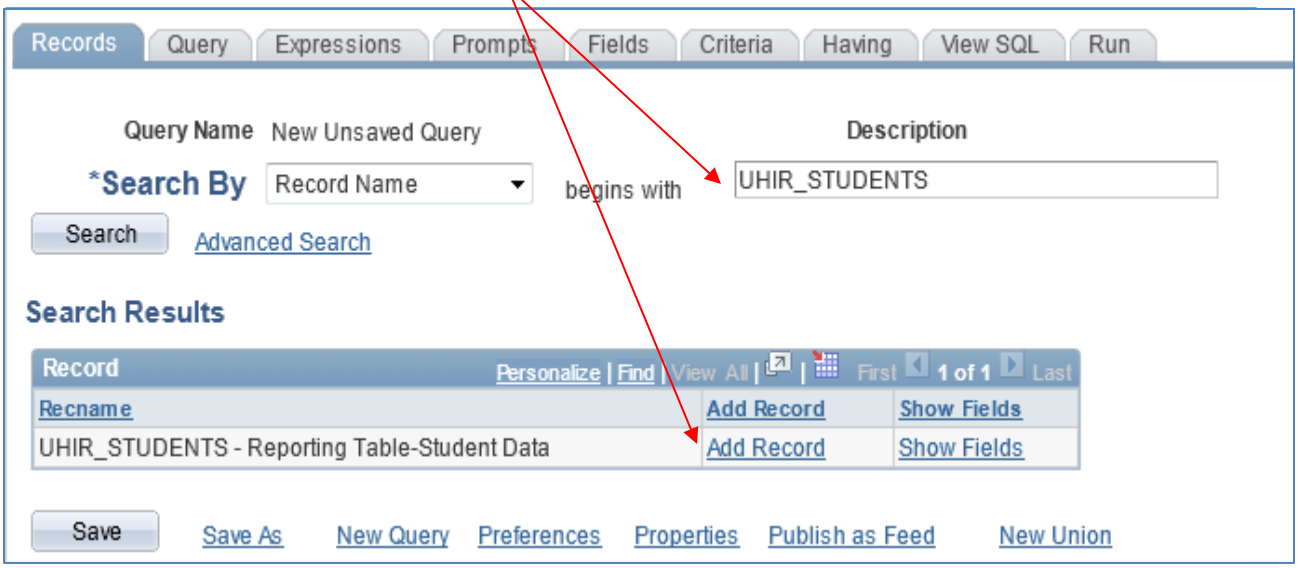
Example *Create a query that lists student's mailing address1 and address2 columns into one combined column*.

Step 1: Navigate to Query Manager

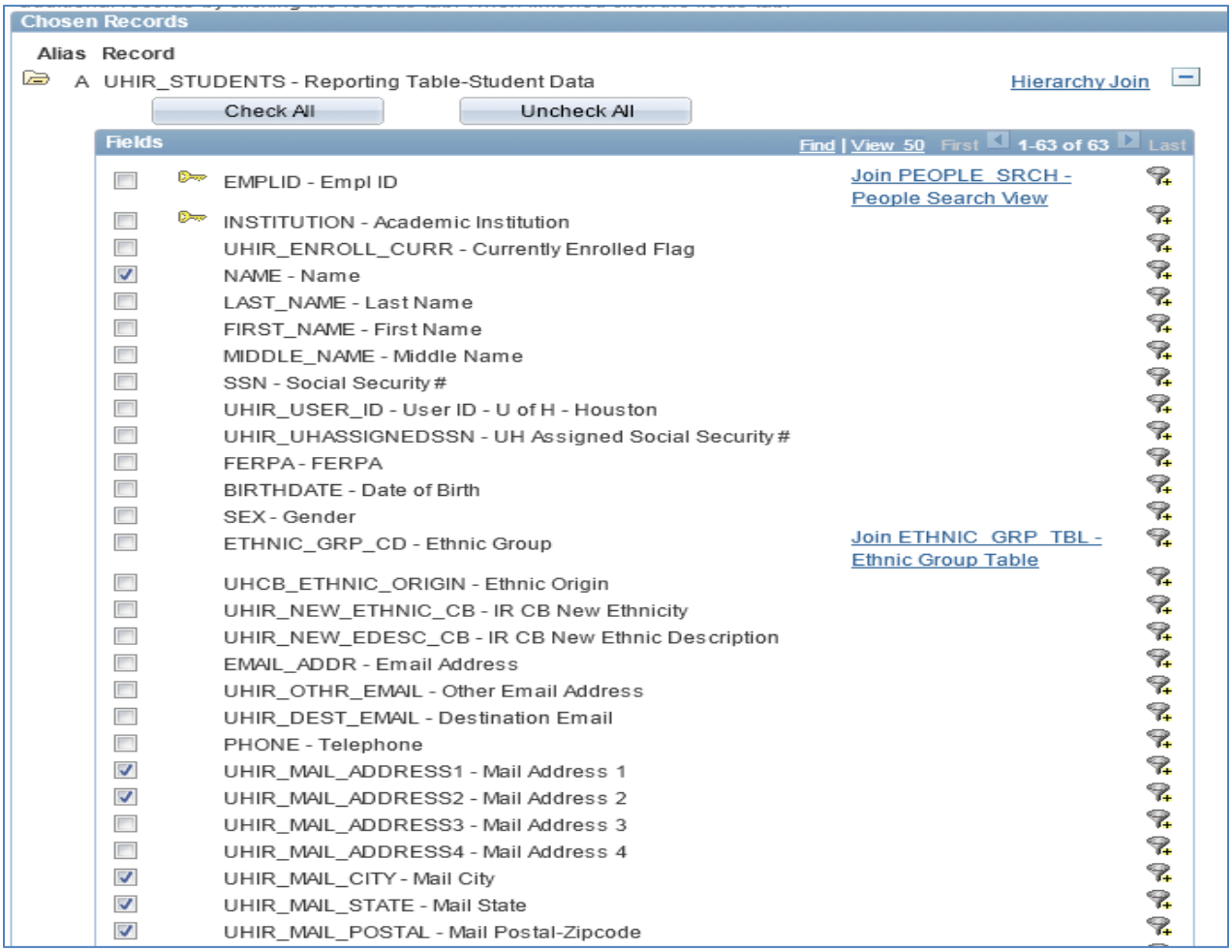
Step 2: Click on Create New Query.



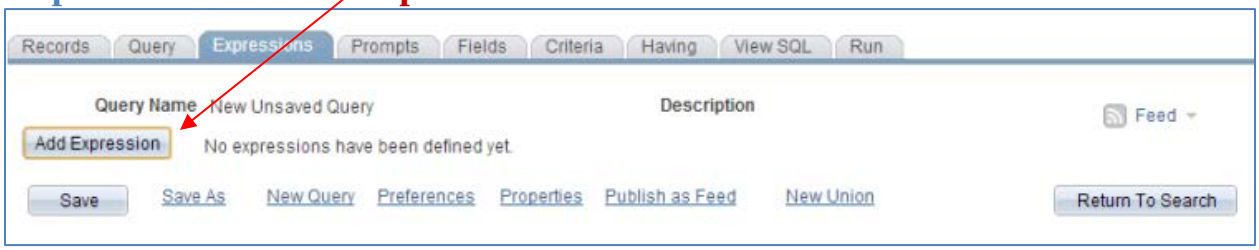
Step 3: Add the UHIR_STUDENTS record.



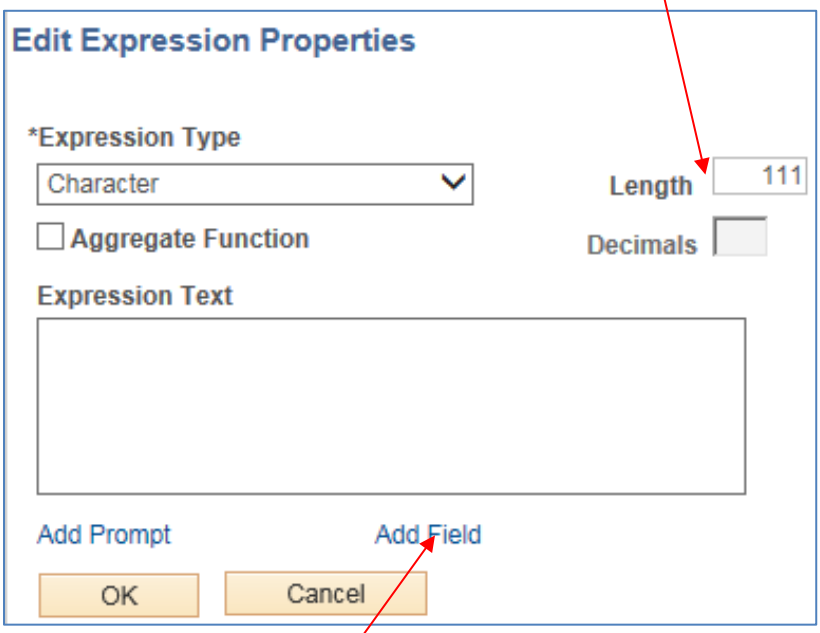
Step 4: Click on the fields to be listed on the report (Name, UHIR_MAIL_ADDRESS1, UHIR_MAIL_ADDRESS2, UHIR_MAIL_CITY, UHIR_MAIL_STATE, UHIR_MAIL_POSTAL)



Step 5: Click on 'Add Expression'.

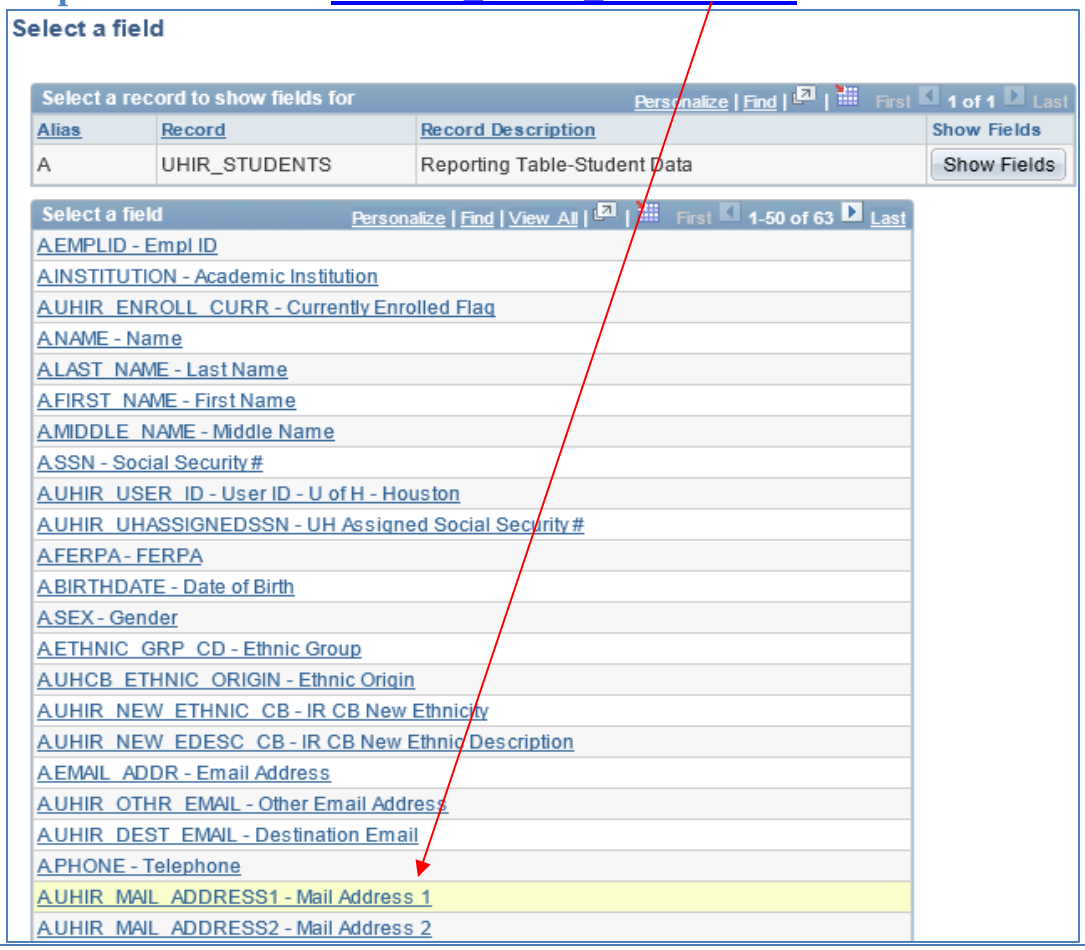


Step 6: Change the length of the field to 111 so that the address field displays 111 characters.

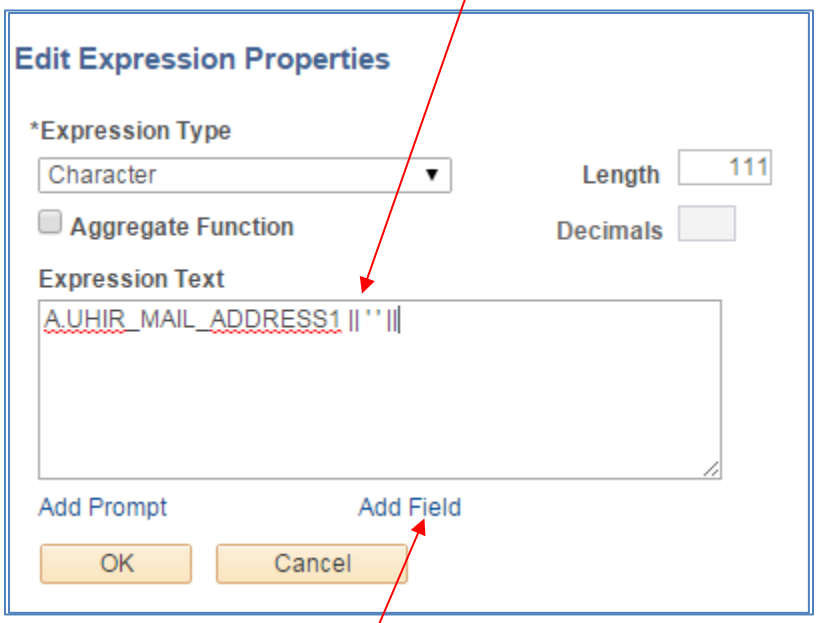


Step 7: Click on Add Field.

Step 8: Click on the A.UHIR_MAIL_ADDRESS1 link.

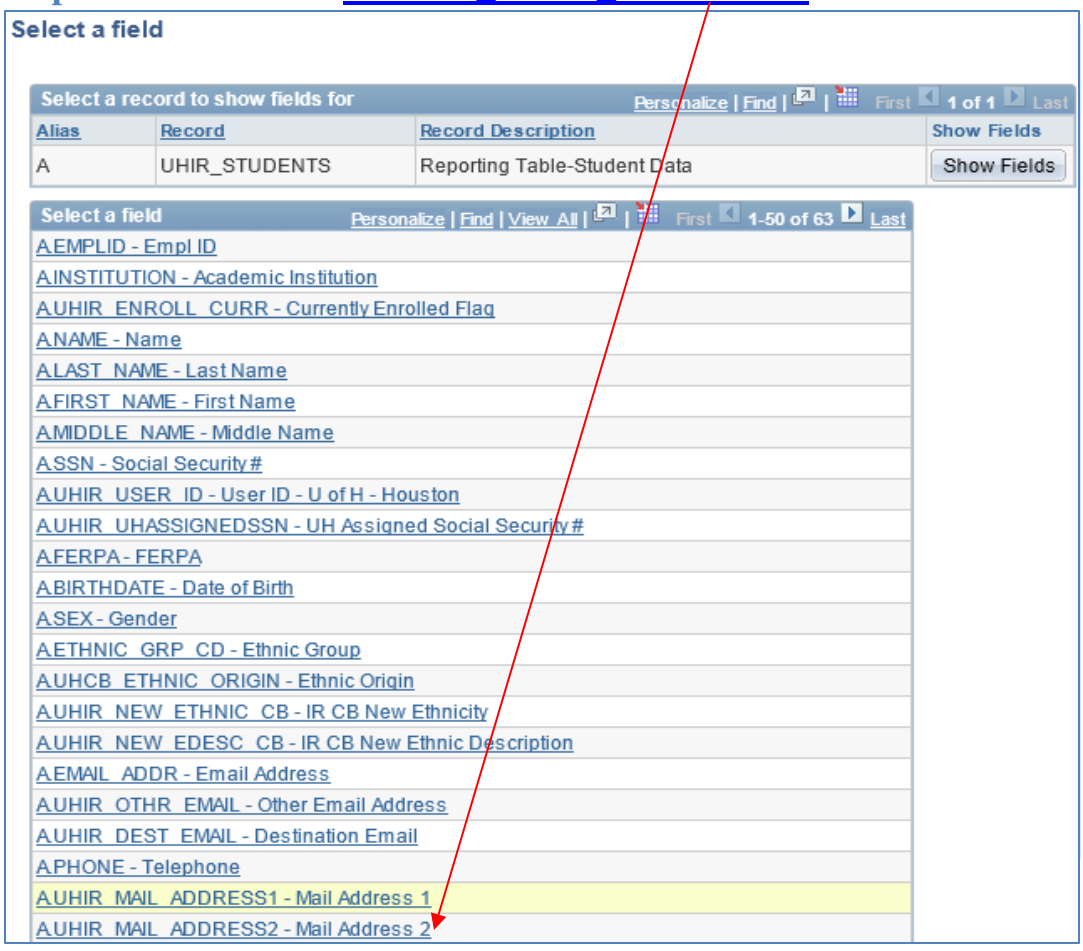


Step 9: Type || ' ' || to add a space between the ADDRESS1 field and the ADDRESS2 field.

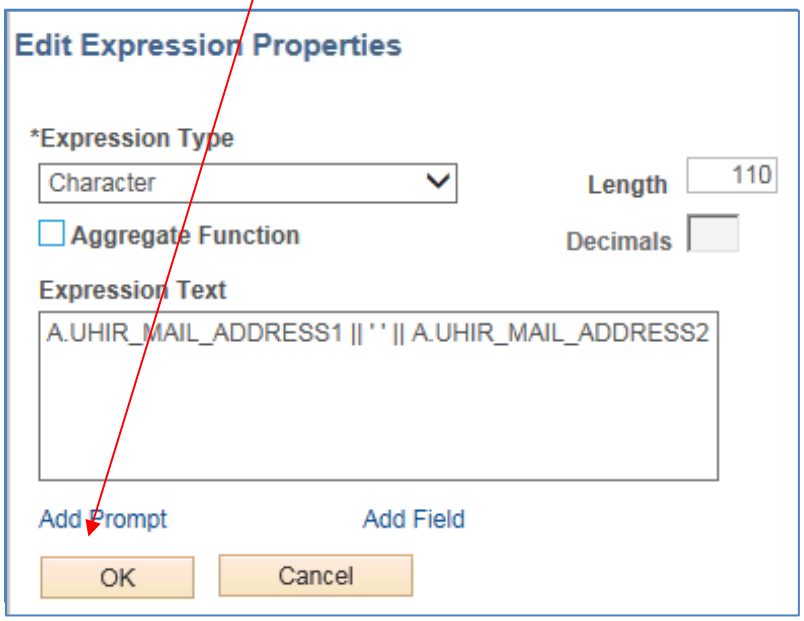


Step 10: Click on Add Field.

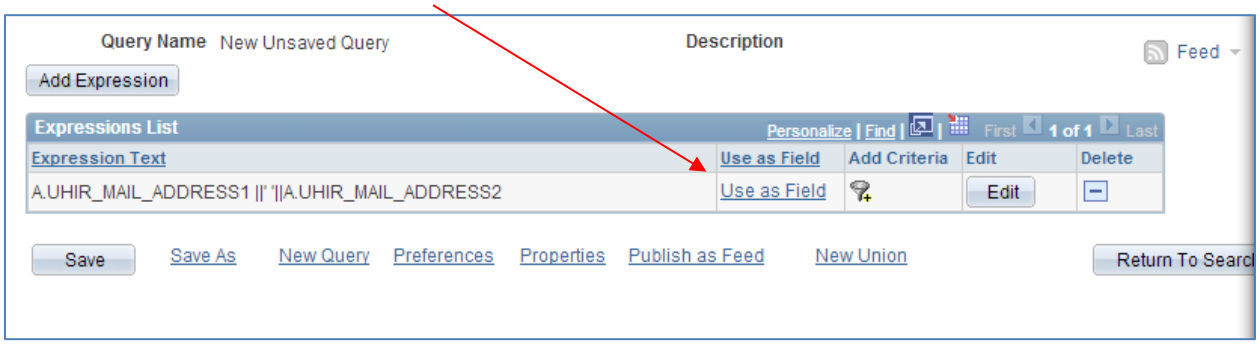
Step 11: Click on the A.UHIR_MAIL_ADDRESS2 link.



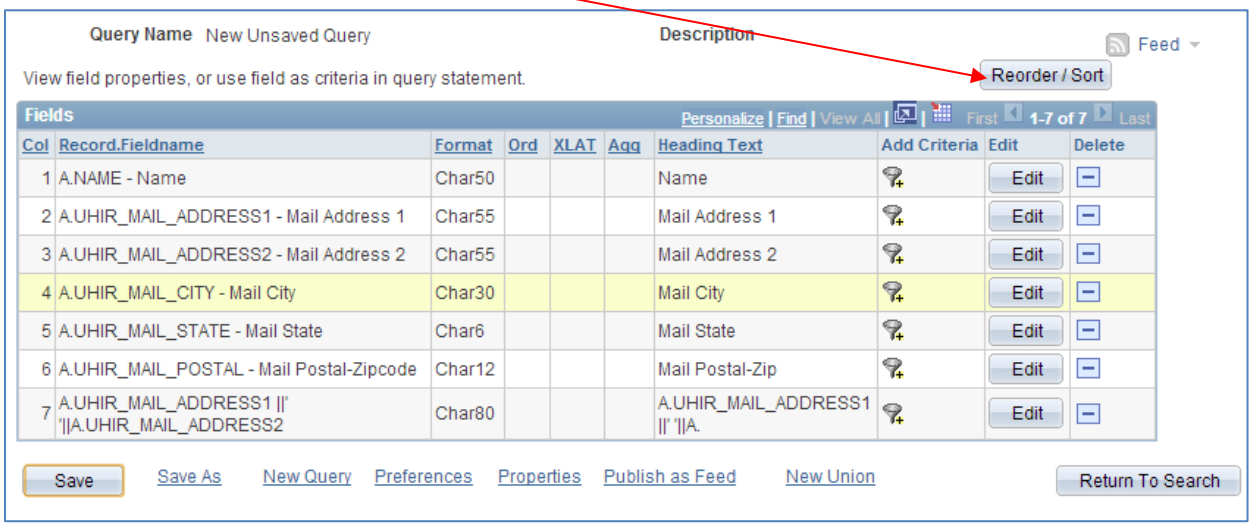
Step 12: Click OK



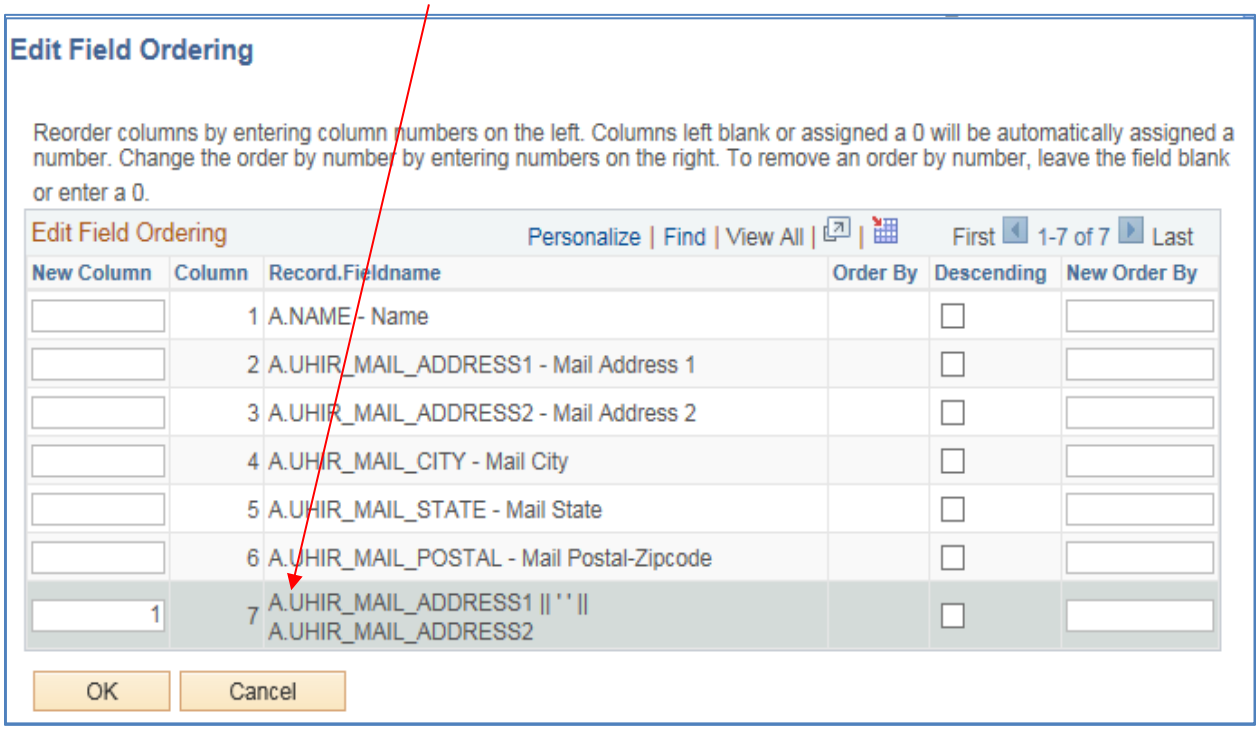
Step 13: Click Use as Field.



Step 14: Click on the 'Reorder/Sort' button to change the order of the columns.



Step 15: Enter the Column Order as listed below.



Edit Field Ordering

Reorder columns by entering column numbers on the left. Columns left blank or assigned a 0 will be automatically assigned a number. Change the order by number by entering numbers on the right. To remove an order by number, leave the field blank or enter a 0.

Edit Field Ordering — Personalize | Find | View All | First 1-7 of 7 Last

New Column	Column	Record.Fieldname	Order By	Descending	New Order By
	1	A.NAME - Name		☐	
	2	A.UHIR_MAIL_ADDRESS1 - Mail Address 1		☐	
	3	A.UHIR_MAIL_ADDRESS2 - Mail Address 2		☐	
	4	A.UHIR_MAIL_CITY - Mail City		☐	
	5	A.UHIR_MAIL_STATE - Mail State		☐	
	6	A.UHIR_MAIL_POSTAL - Mail Postal-Zipcode		☐	
1	7	A.UHIR_MAIL_ADDRESS1 || ' ' || A.UHIR_MAIL_ADDRESS2		☐	

OK Cancel

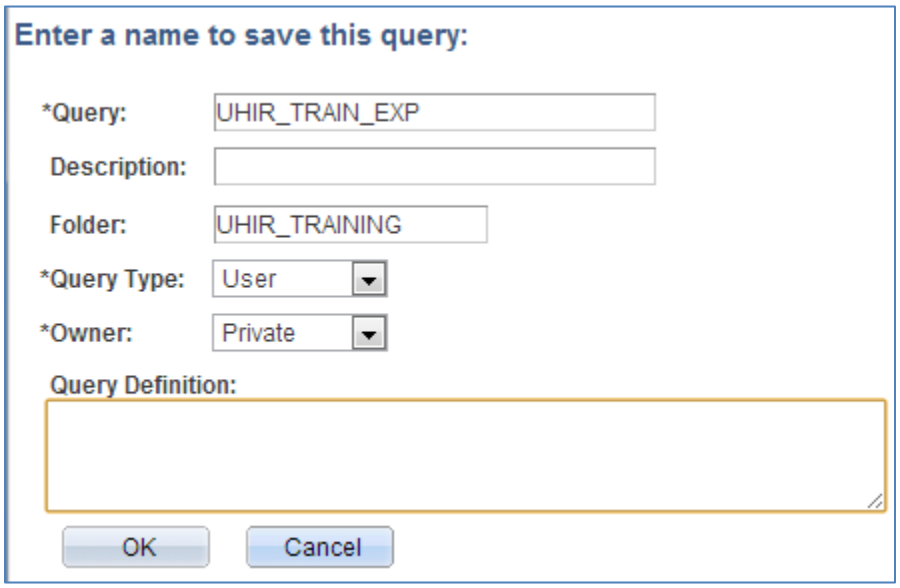
Step 16: Click on the 'Run' tab to view the output in query.

Query Results

View All | Rerun Query | Download to Excel | Download to XML — First 1-100 of 7877 Last

	A.UHIR_MAIL_ADDRESS1 || ' ' ||	Name	Mail Address 1	Mail Address 2	Mail City	Mail State	Mail Postal-Zip
1	827 West 17th Street	Johnson,Yolanda	827 West 17th Street		Houston	TX	77008-3527
2	22526 Ganado Creek Ct	Nguyen,Nhu D	22526 Ganado Creek Ct		Katy	TX	77449-4844
3	20815 Westgreen Court	Cassady,Daniel Wade	20815 Westgreen Court		Katy	TX	77450-4131
4	4735 Stoney Point Court	Hall,Melody Joyce	4735 Stoney Point Court		Sugar Land	TX	77479-5203
5	405 Hampshire St	Mount,Lora S.	405 Hampshire St		Victoria	TX	77904-2277
6	105 Shiloh Drive	Yur,Amy Elizabeth	105 Shiloh Drive		Victoria	TX	77904-2434
7	Apt A 1524 Nantucket Dr	Cheraghpour,Nima	Apt A	1524 Nantucket Dr	Houston	TX	77057-1945
8	Hwy 111 N PO Box 821	Carroll,Robert Russell	Hwy 111 N	PO Box 821	Edna	TX	77957-0821
9	2121 County Road 235	Biegel,Heather Michelle	2121 County Road 235		Alvin	TX	77511-6746
10	6813 Savannah Lane	Gendreau,Kelly Lynn	6813 Savannah Lane		Fort Worth	TX	76132-3731

Step 17: Save the query (UHIR_TRAIN_EXP) as a Private Query. ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.



Chapter 10: In-Class Exercise (Adding Expressions)

Query 1

Create a query listing currently enrolled UH-Main students and their combined Mail City, State and Zip. For the combined column, place a comma between the City and the State and a space between State and Zip. Select fields: Name, Mail Address1, Mail Address2, Mail City, Mail State, Mail Postal-Zip, and the concatenated expression. Sort by Name. Save the query as UHIR_TRAIN_EXP1. **ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.**

Query 2

Create a query listing UH-Main students admitted in term 1910 and their total SAT English plus Math scores, exclude students where the total SAT English plus SAT Math equals zero. List EMPLID, UHIR_SAT_ENG, UHIR_SAT_MATH, and the total English plus Math score. Sort by the EMPLID field in ascending order. (Hint: Expression Type is Number with Length of 7 and Decimals of 2). Save the query as UHIR_TRAIN_EXP2. **ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.**

Query 3

Using previously created query UHIR_TRAIN_EXP1, add a column that indicates if a marketing letter should be mailed to the student based on the student's FERPA preference. If FERPA is flagged as Yes for the student populate the column data to say "Include in Statistics, BUT DO NOT MAIL", otherwise populate the column data to say "Include in Statistics, OK to Mail". Add the FERPA field and add the new expression column to display in the output at the end of the prior selected columns, the new expression should have custom heading "Mailing Instructions". Sort the output by the FERPA field in descending order, then sort by Name. Also add criteria where Mail Postal-Zipcode starts with 7737. Save the query as UHIR_TRAIN_EXP3. **ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.**

HINT , use SQL Function DECODE in the expression (if needed, more detail is available in the Appendix):

DECODE(x, val1, result1[,val2, result2, …, valN, result] [, default])

Example(s):

• DECODE(A.COLOR, 'R', 'Red', 'G', 'Green', 'B', 'Blue', 'Unknown')

- If A.COLOR is "R" then DECODE returns "Red"
- If A.COLOR is "G" then DECODE returns "Green"
- If A.COLOR is Blank then DECODE returns "Unknown"

CHAPTER 11 – Joining Multiple Tables

Overview

Query Manager enables you to create queries that include multiple-table *joins.* Joins retrieve data from more than one table, presenting the data as if it came from one table. PeopleSoft Query links the tables, based on common columns, and links the rows on the two tables by common values in the shared columns.
Joins are what make relational databases relational. Using joins, you define relationships among fields when you query the records, not when you create the records. Because PeopleSoft records are highly normalized (they each describe one kind of entity), you can easily use Query Manager to create joins.

Example

Create a query that joins two records to list the 'LAW' students that are enrolled at UH-Main, as of census, in the fall 2007 semester. The query should include the degree information if the students received a degree in 2007.

Understanding Join Categories and Join Types

The procedure for joining tables differs depending on how the tables that are being joined are related to each other. Query Manager recognizes three categories of joins: record hierarchy, related record, and any record. Query Manager also recognized two types of joins based on how the tables are related to each other, they are: inner joins and outer joins. **Key to creating queries with joins is identifying the appropriate data elements (fields) that must be connected in the query**.

Categories of Joins:

The categories of joins focus on what **relationships** exist between records that will be connected in a query. The categories of joins include:

- Record Hierarchy Join
- Related Record Join
- Any Record Join

Types of Joins:

The types of joins focus on **how** records are connected in a query. The types of joins include:

- Inner Join (aka Standard Join)
- Outer Join (aka Left Join and Left Outer Join)

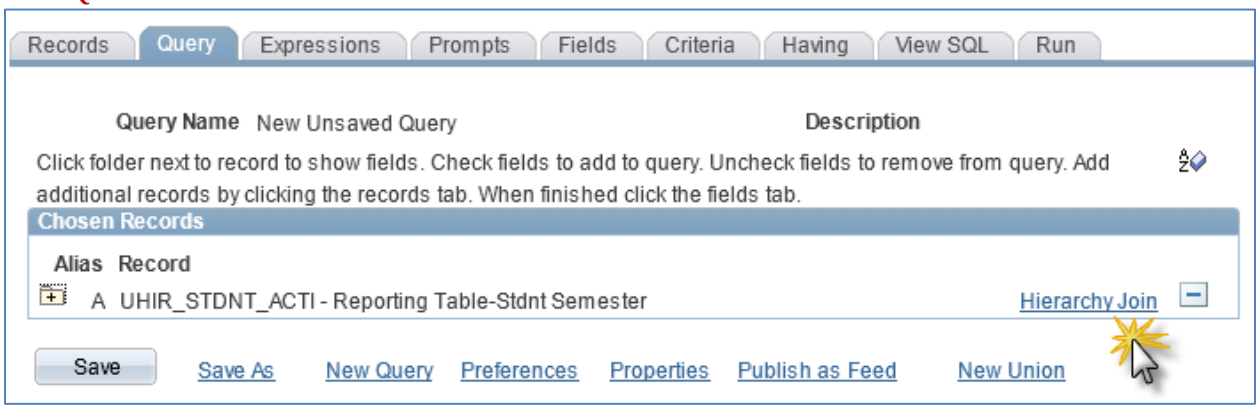
Record Hierarchy Join

Record Hierarchy Joins connects a parent table to a child table. A child table is a record that uses all the same key fields as its parent plus one or more additional keys. For example the UHIR_STUDENTS record has a parent/child relationship with the UHIR_STDNT_ACTIV (Activities) record.

To create Record Hierarchy joins, click on the '**Hierarchy Join**' link next to the field having a hierarchal relationship in the Query tab. All of the records that have a parent/child relationship with the base record you've selected will be listed (the parent/grandparent of that record will be listed as well as the children/grandchildren of that record will be listed). Then click on the link of the record you want to perform the hierarchal join on.

Note: For the record hierarchy relationships to show up in the Query Tool, the developers have to set up that relationship when designing the records. If a developer does not set up the relationship, it will not show up in the Query Tool.

Also note: When you create a Record Hierarchy join, PeopleSoft Query automatically adds an "Inner Join" criteria between the tables on key fields. The hierarchy criteria is embedded within the SQL statement generated by the query; you cannot make changes to the hierarchy criteria and must click on the SQL tab to see the embedded criteria.



Related Record Join

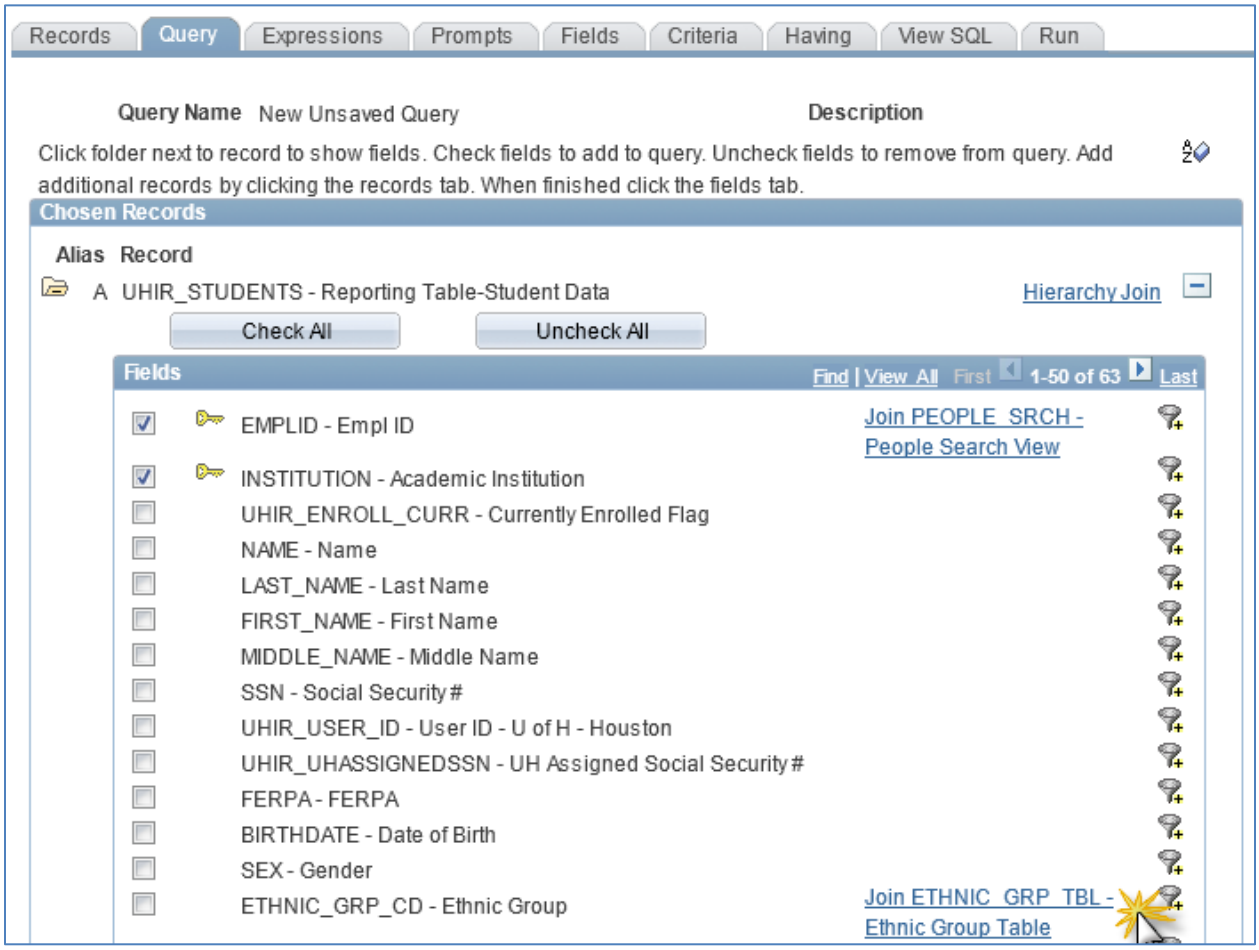
Related Record joins connects records that have been predefined by the developer to show a relationship with each other based on a specific field in the table. Those predefined relationships for fields are indicated in the Query Tool by a link on the right hand side of the field in the Query tab.

Related Record join functionality is very useful in quickly identifying and incorporating additional fields of data from unknown master tables into the query definition.

For example, if building a query using the UHIR_STUDENTS record that queries students ethnicity codes you will note the base record has field 'ETHNIC_GRP_CD' which can be used in the query. However the base record only provides the code for the ethnicity, it does not provide a description of

what the code means. From the Query tab, if you look to the right of ETHNIC_GRP_CD field you will see a link to the field's Related Record which is the 'ETHNIC_GRP_TBL' table. Clicking on this link will allow you to add the ETHNIC_GRP_TBL record to the query and select the description field related to the code so that when run, the query displays the ethnicity code as well as what the code means.

To create Related Record joins, click on the record hyperlink next to the field having a related record relationship in the Query tab. Related Record join functionality allows the user to specify if they want the additional table to be added as either an Inner join or Outer Join. However **note**: the added criteria is embedded within the SQL statement generated by the query; you cannot make changes to the related record criteria and must click on the SQL tab to see the embedded criteria.



Any Record Join *(and Cartesian Joins)*

As useful as Record Hierarchy and Related Record joins are, most queries with joins are actually written using Any Record joins. Many joins are needed where the developer has not created a predefined relationship between the data elements of the records needed in the query, when those predefined relationships are not available users must create Any Record joins.

Any Record Joins are simply joins you create adhoc between two records without using predefined relationships indicated by record hierarchy or related records. It's literally "any" join you create between two records on their appropriate fields.

In creating an Any Join you have to be very careful. **Best practice dictates the query writer should do some research first to gain clarity of how the data lies within the two tables individually so that proper joins between related data elements can be created when creating or modifying the query**. Improperly creating Any Join criterion can result in incorrect data being displayed and mistaken as good data; as well as cause database performance issues from Cartesian joins.

A Cartesian join is a join of every row of one table to every row of another table. This normally happens when no matching join columns are specified. For example, if table A with 100 rows is joined with table B with 1000 rows, a cartesian join will return 100,000 rows. Something to be avoided!

Note: A query must have at least (N-1) join conditions to prevent a cartesian product, where N is the number of tables in the query.

In the section of this manual discussing Query Preferences, the "Enable Auto Join" functionality was presented. Query writers were encouraged to keep this setting enabled so that the Query Tool does the preliminary work for you to determine what some of the appropriate fields are to match data elements between the two records.

Inner Join

Inner Joins are also referred to as *Standard Joins*. An Inner Join is the type of join where the values in the field joined on must exist in both tables for the row of data to be displayed when the query is run. If the value exists in one table but does not exist in the other table for that field then the row will not be displayed as output when the query is run.

Outer Join

Outer Joins are also referred to as *Left Joins* and *Left Outer Joins*. An Outer Join is the type of join where a row of data from one of the participating tables (the base table) appear in the query output even if a matching row of data doesn't exist in the other table (the supplemental table) based on the field(s) used in the join.

Outer Joins allow you to see all data in the base record regardless if a corresponding row of data exists in the supplemental record; and if a row of data exists in both the base record and supplemental record then you will see that additional data from the supplemental record in the output.

Creating a Query with Multiple Records

Given most queries with joins use Any Record joins, the chapter example below utilizes the Any Record functionality to illustrate creating queries with multiple tables.

Step 1: By now you should be able to create a simple query that selects from one table. Create a query that lists the EMPLID and ACAD_CAREER for all students enrolled:

1) As of census
2) At UH-Main
3) During the fall 2007 semester
4) ACAD_CAREER = 'LAW'

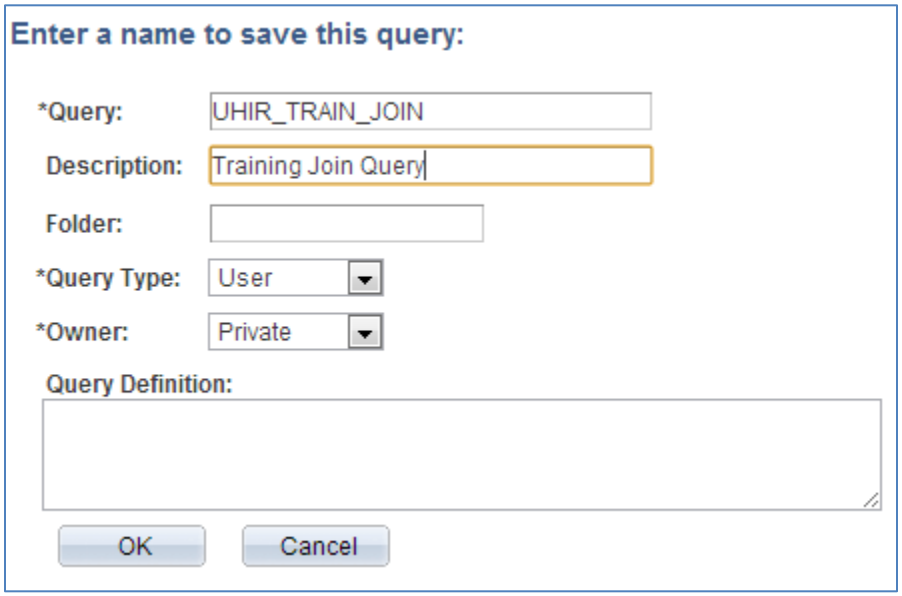
Step 2: Run the query and write down the number of records returned.

View All | Rerun Query | Download to Excel | Download to XML

First 1-100 of 1052 Last

	ID	Career
1	0565437	LAW
2	0566142	LAW
3	0569054	LAW
4	0633307	LAW
5	0634223	LAW
6	0634729	LAW
7	0634832	LAW
8	0659690	LAW
9	0701127	LAW
10	0747222	LAW
11	0784640	LAW
12	0817026	LAW
13	0817074	LAW
14	0817297	LAW
15	0817484	LAW
16	0817822	LAW

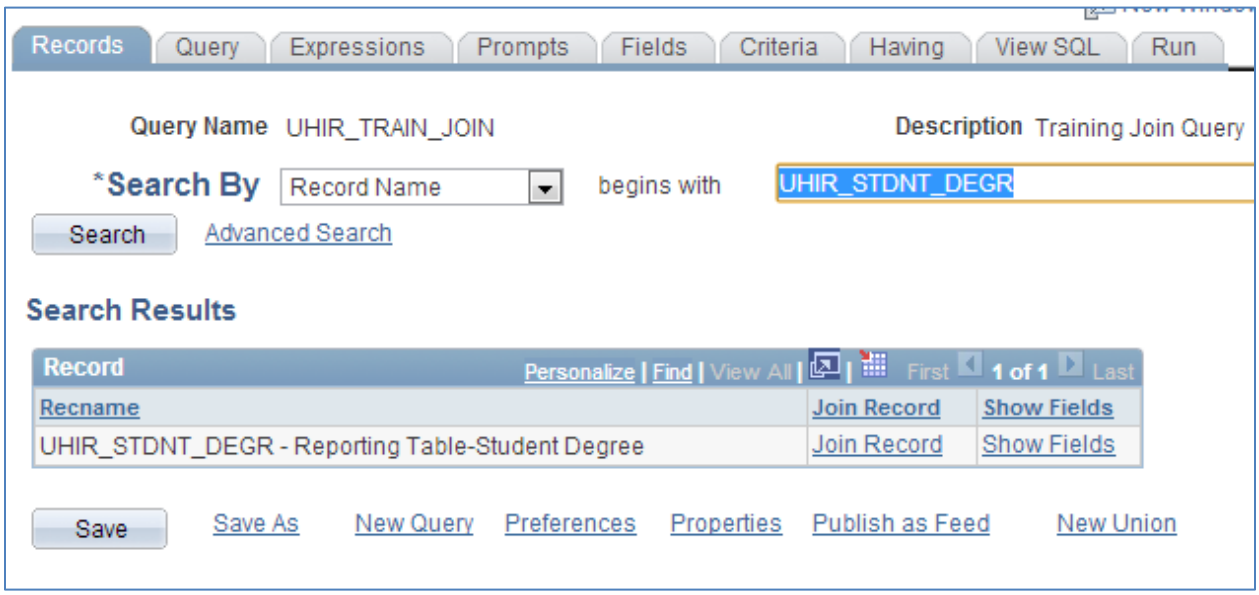
Step 3: Save the query as a private query (UHIR_TRAIN_JOIN).



Step 4: Click on the Records tab.

Step 5: Enter UHIR_STDNT_DEGR in the 'begins with' edit box.

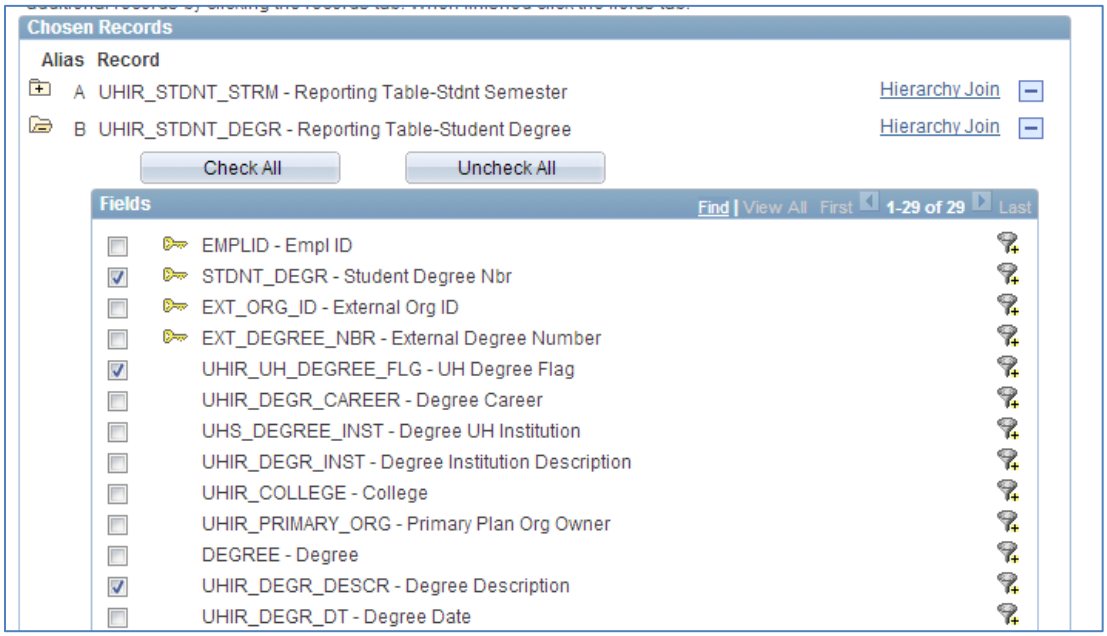
Step 6: Click the Join Record link.



Step 7: Click the UHIR_STDNT_STRM link.

Step 8: Click the 'Add Criteria' button.

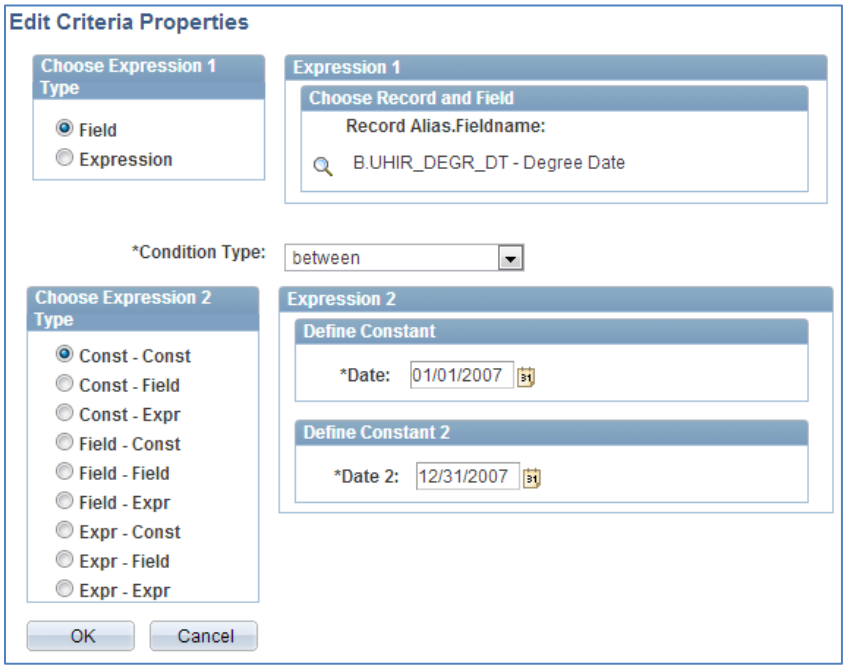
Step 9: Click the below fields to be displayed on the report.



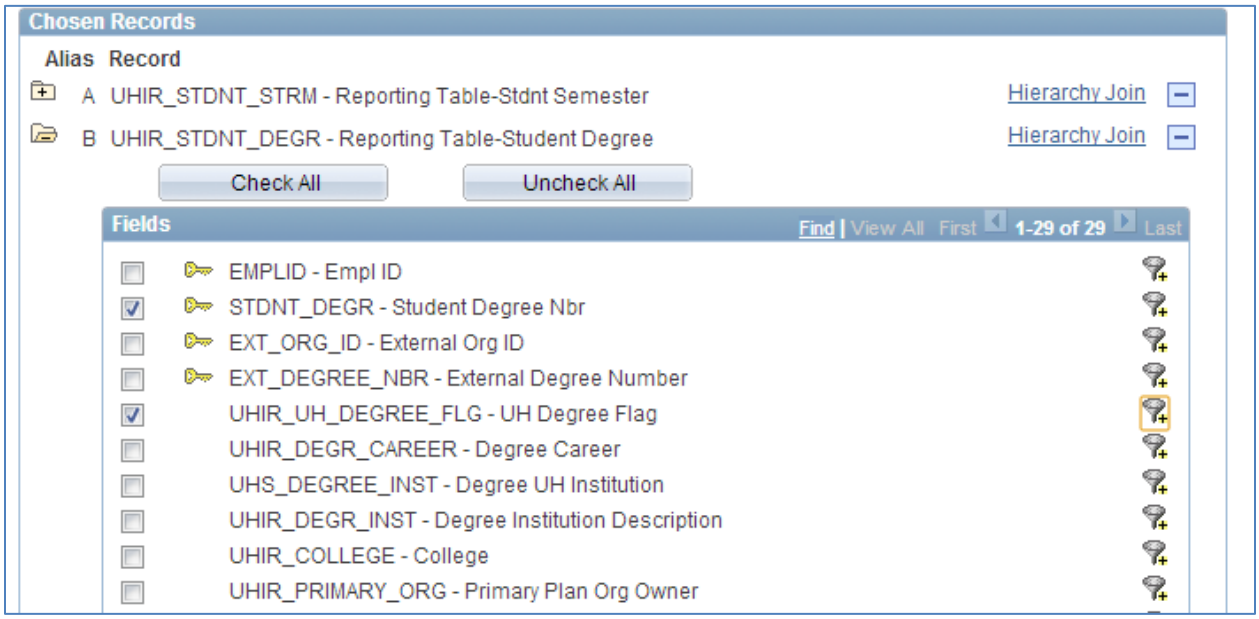
Step 10: Click the 'Use As Criteria' funnel next the UHIR_DEGR_DT field.

Step 11: Select the 'between' Condition Type.

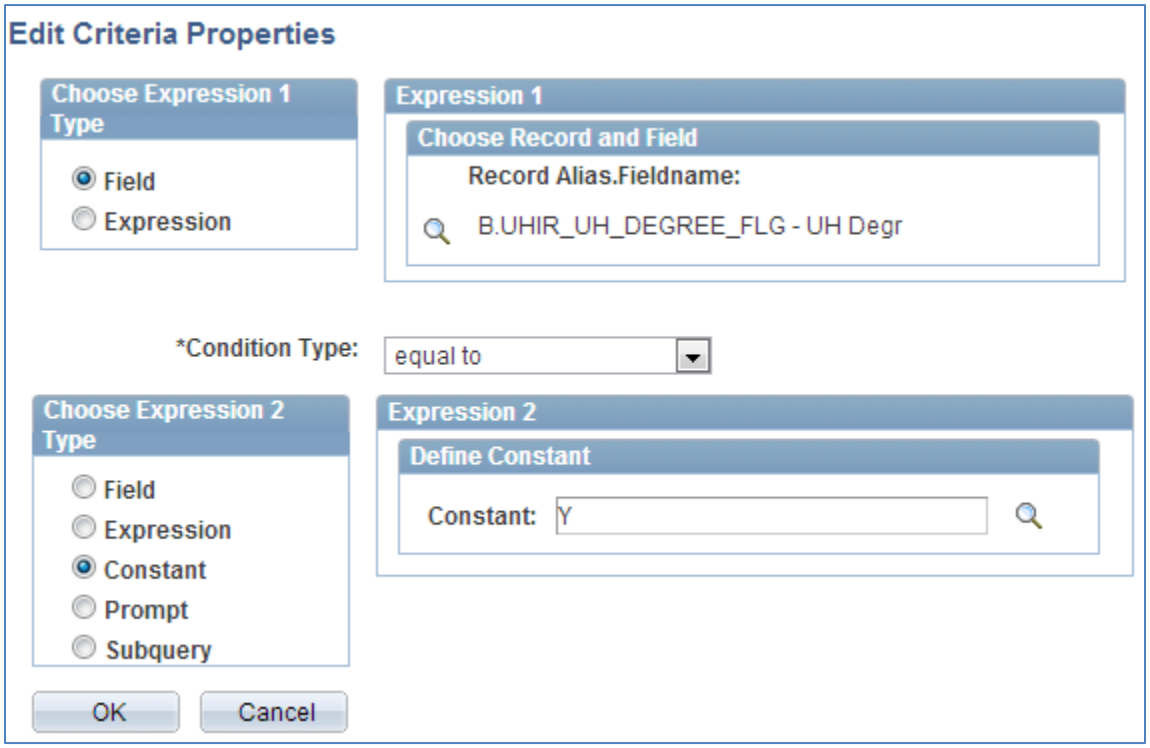
Step 12: Enter the Constant (1/1/2007) and Constant 2 (12/31/2007) values and click 'OK'.



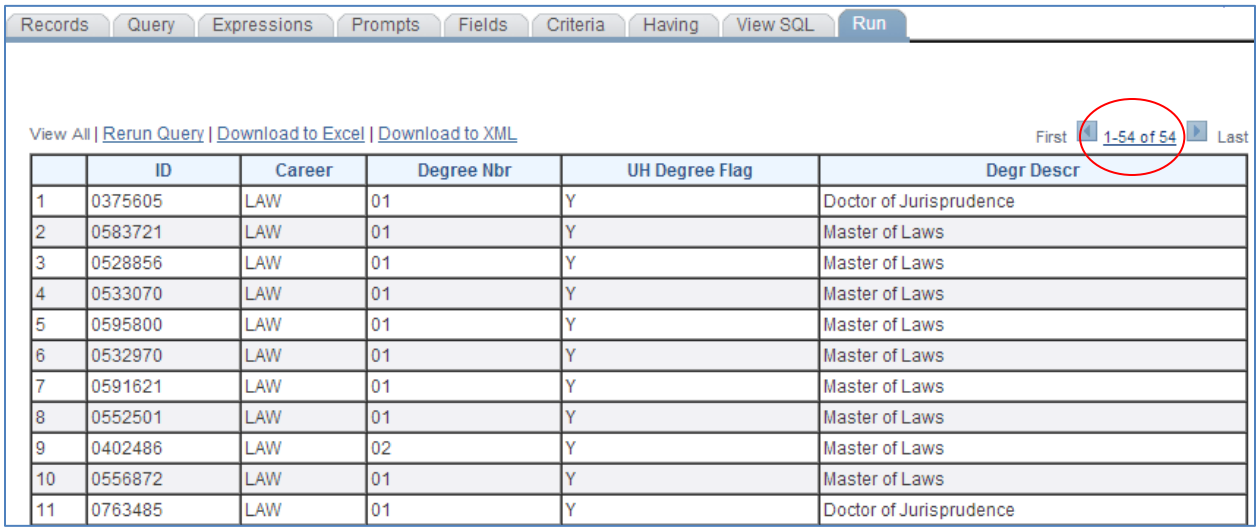
Step 13: Click the 'Use As Criteria' funnel next to the UHIR_UH_DEGREE_FLG.



Step 14: Enter 'Y' in the Constant edit box.



Step 15: Run the query by clicking on the 'Run' tab.



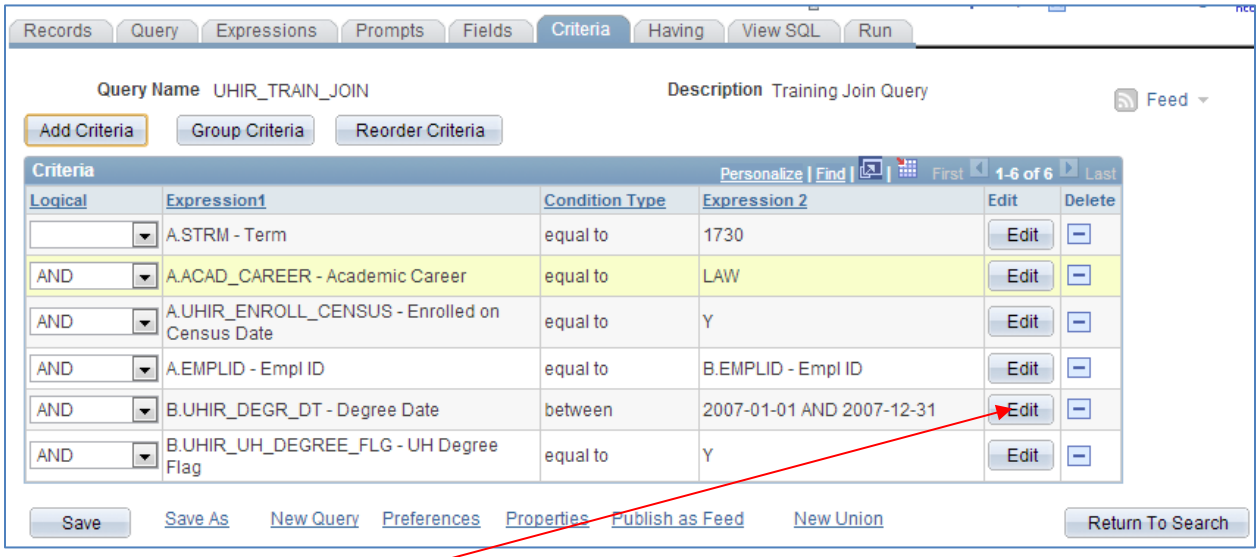
	ID	Career	Degree Nbr	UH Degree Flag	Degr Descr
1	0375605	LAW	01	Y	Doctor of Jurisprudence
2	0583721	LAW	01	Y	Master of Laws
3	0528856	LAW	01	Y	Master of Laws
4	0533070	LAW	01	Y	Master of Laws
5	0595800	LAW	01	Y	Master of Laws
6	0532970	LAW	01	Y	Master of Laws
7	0591621	LAW	01	Y	Master of Laws
8	0552501	LAW	01	Y	Master of Laws
9	0402486	LAW	02	Y	Master of Laws
10	0556872	LAW	01	Y	Master of Laws
11	0763485	LAW	01	Y	Doctor of Jurisprudence

Results Note:

You will notice that not all Law students are returned in this example. This means that not all Law students that were previously displayed (i.e.1000+), no longer show up on the report. An outer join is needed to display all records, even if the student has not earned a degree.

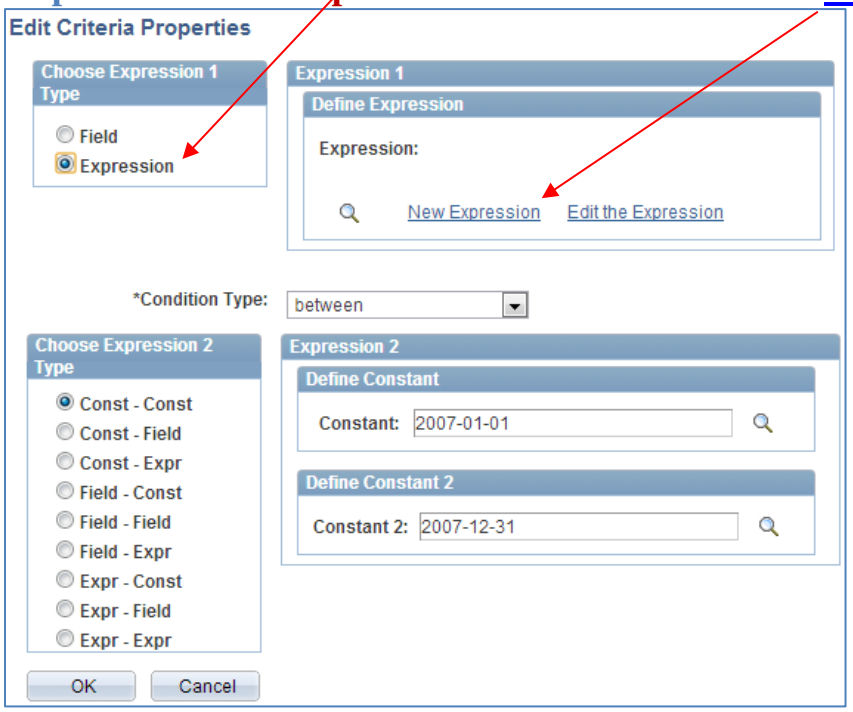
Creating an Outer Join

The outer join returns all records that meet the criteria on the outer record, even if records are not found on non-outer records. To create an outer join, indicate the outer join record by adding a (+) next to each of the criteria arguments in the supplemental record (i.e. the record missing the rows).

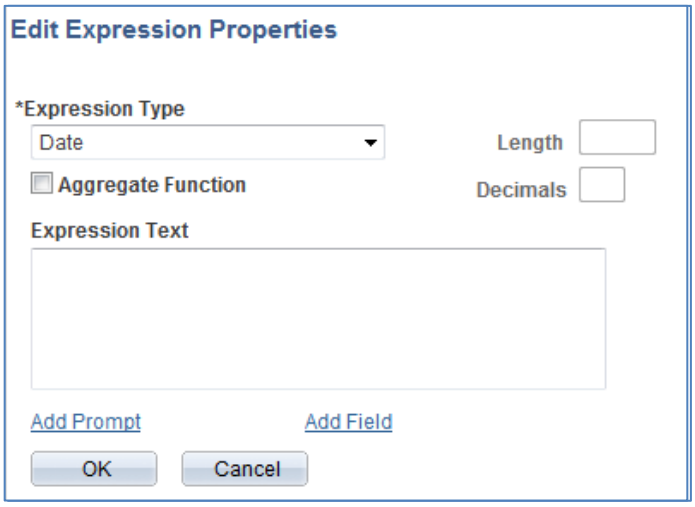


Step 1: Click the 'Edit' button next to the B.UHIR_DEGR_DT field.

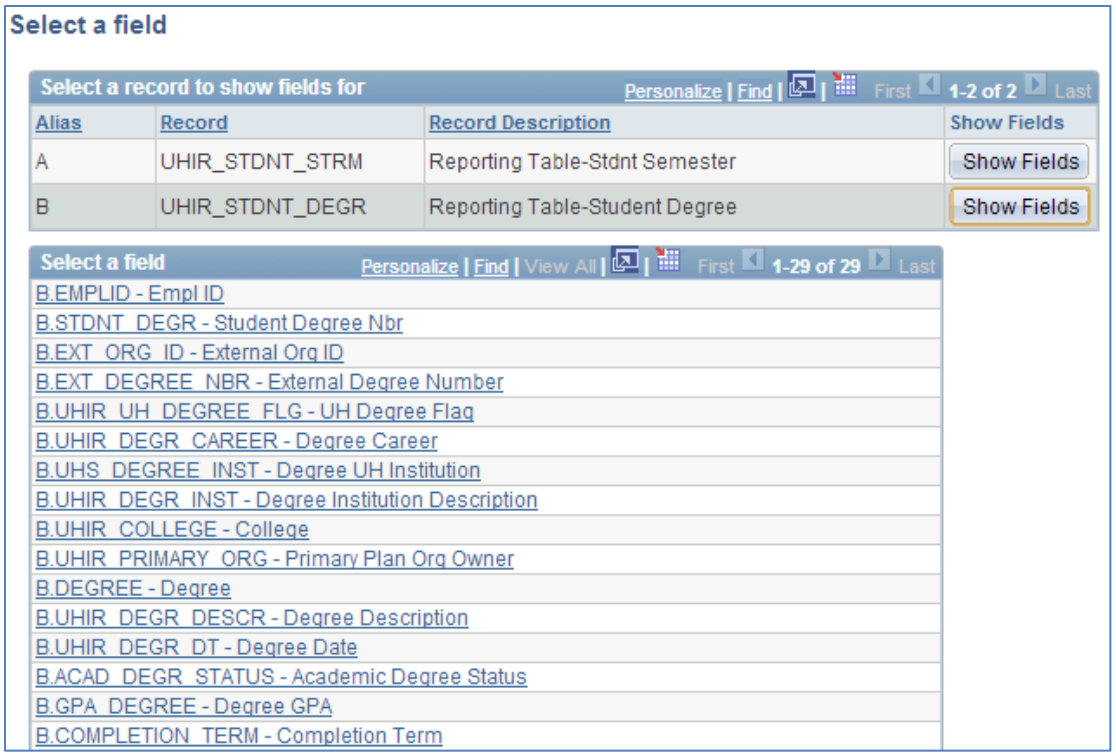
Step 2: Click the 'Expression' radio button and the New Expression link.



Step 3: Change the Expression Type to 'Date' and Click the Add Field link.

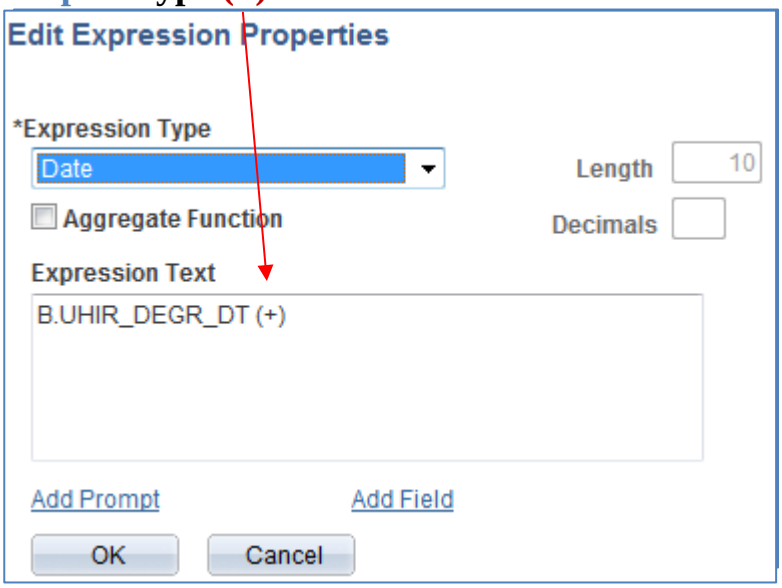


Step 4: Click on the 'Show Fields' button next to the UHIR_STDNT_DEGR record.

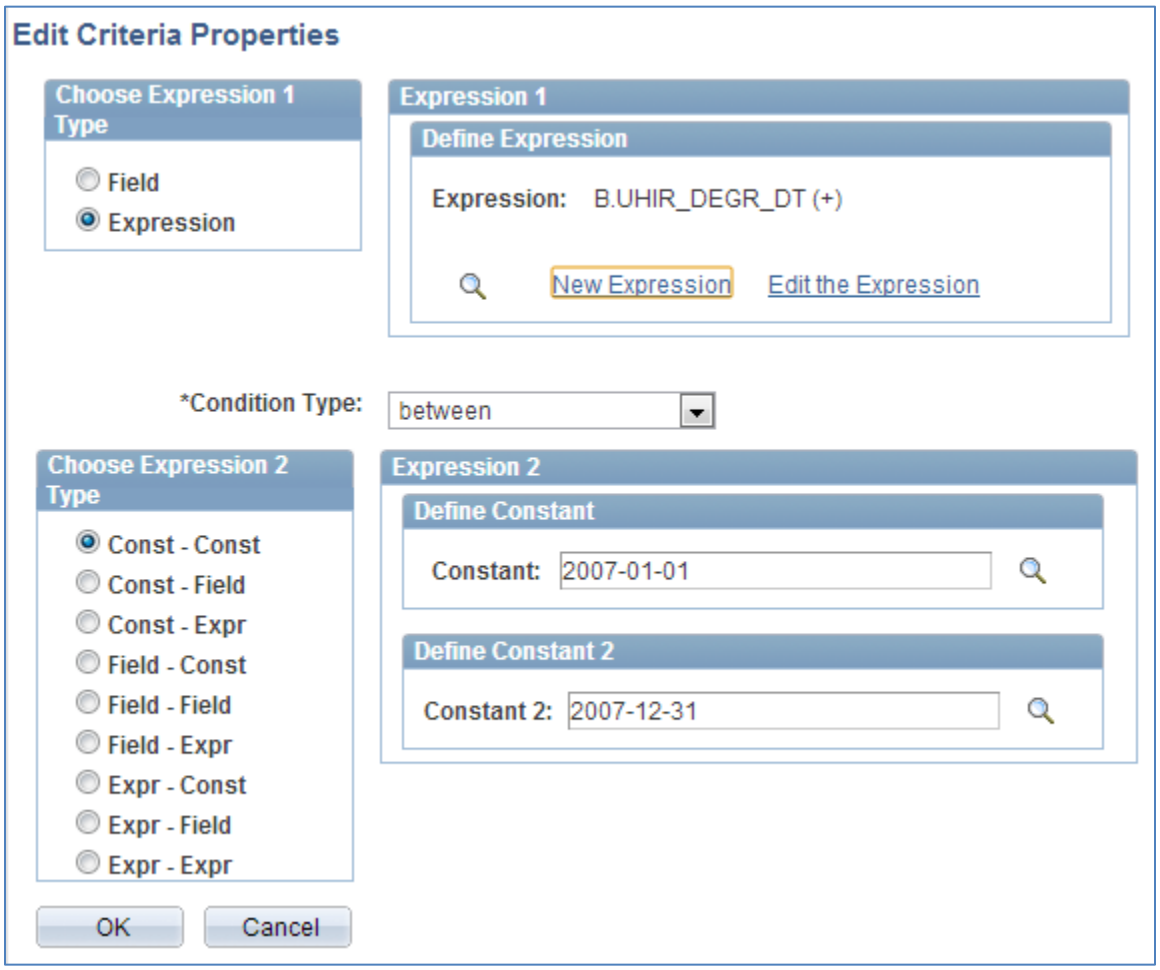


Step 5: Click on the UHIR_DEGR_DT link.

Step 6: Type (+) after the field and click 'OK'. ***This is interpreted as an outer join.***

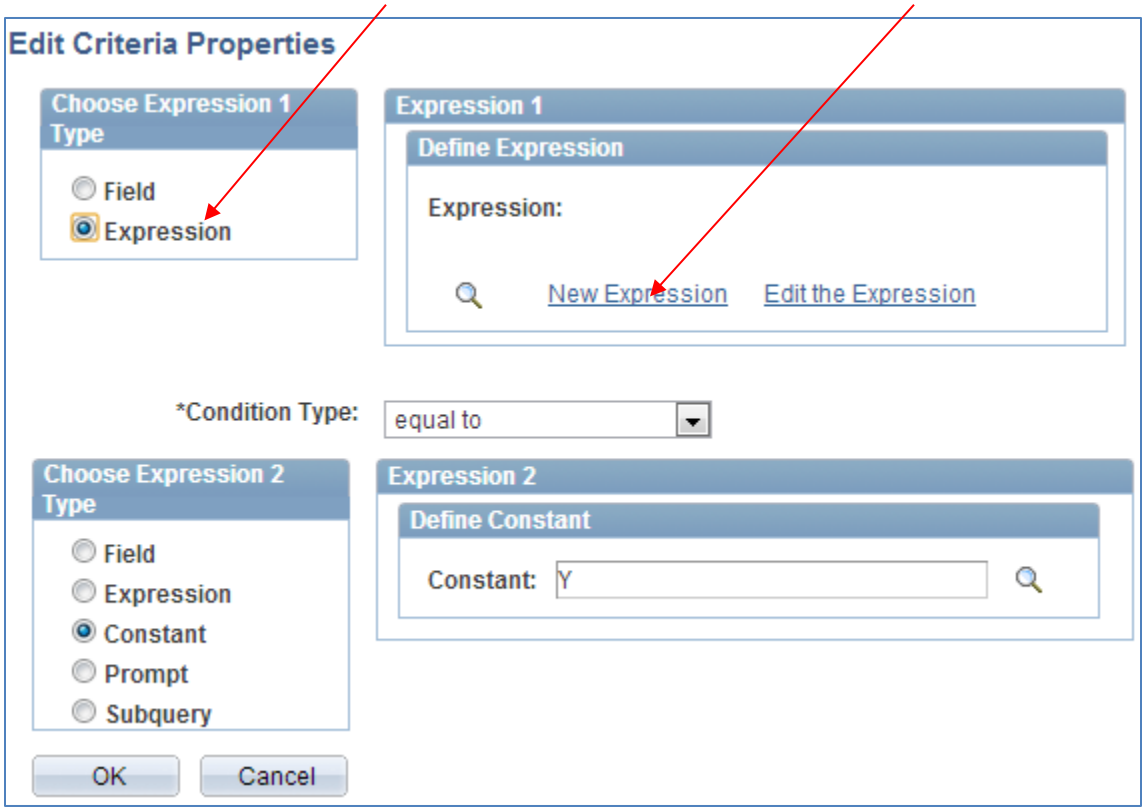


Step 7: Click 'OK'.

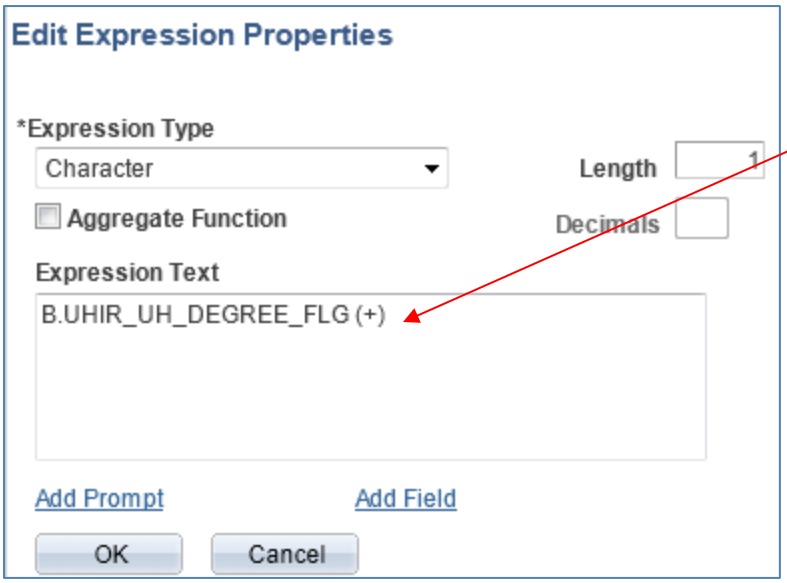


Step 8: Click the 'Edit' button next to the B.UHIR_UH_DEGR_FLG field.

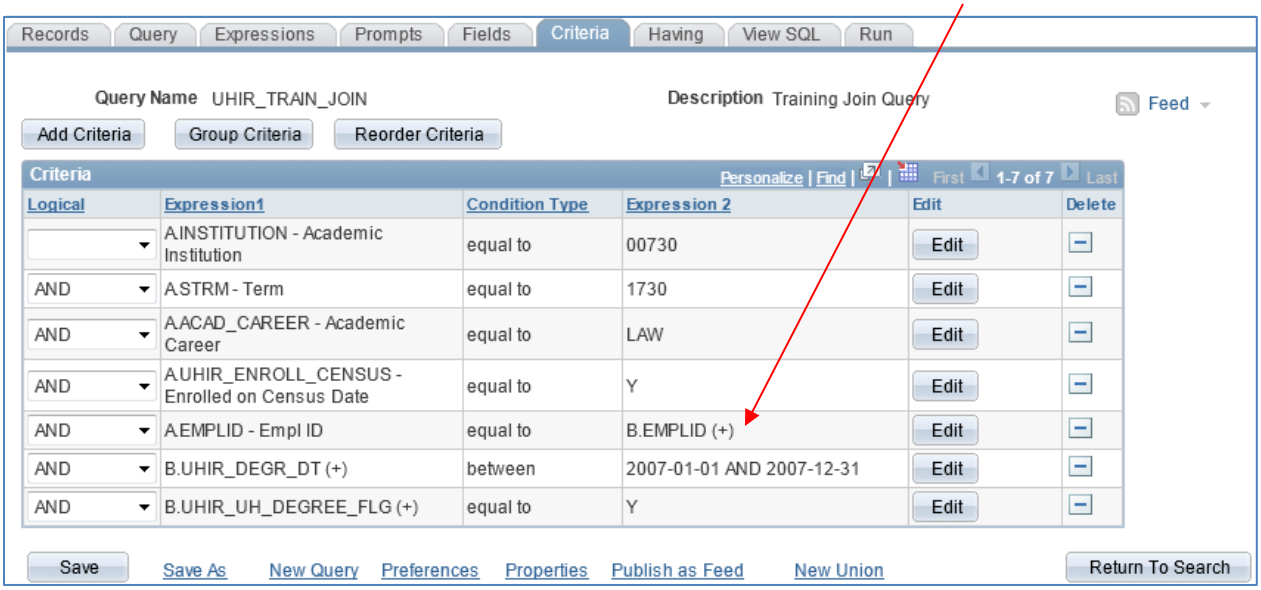
Step 9: Click the 'Expression' radio button and the New Expression link.



Step 10: Click the Add Field link and select B.UHIR_DEGREE_FLG. Type (+) after the field and click 'OK'. *This is interpreted as an outer join*.

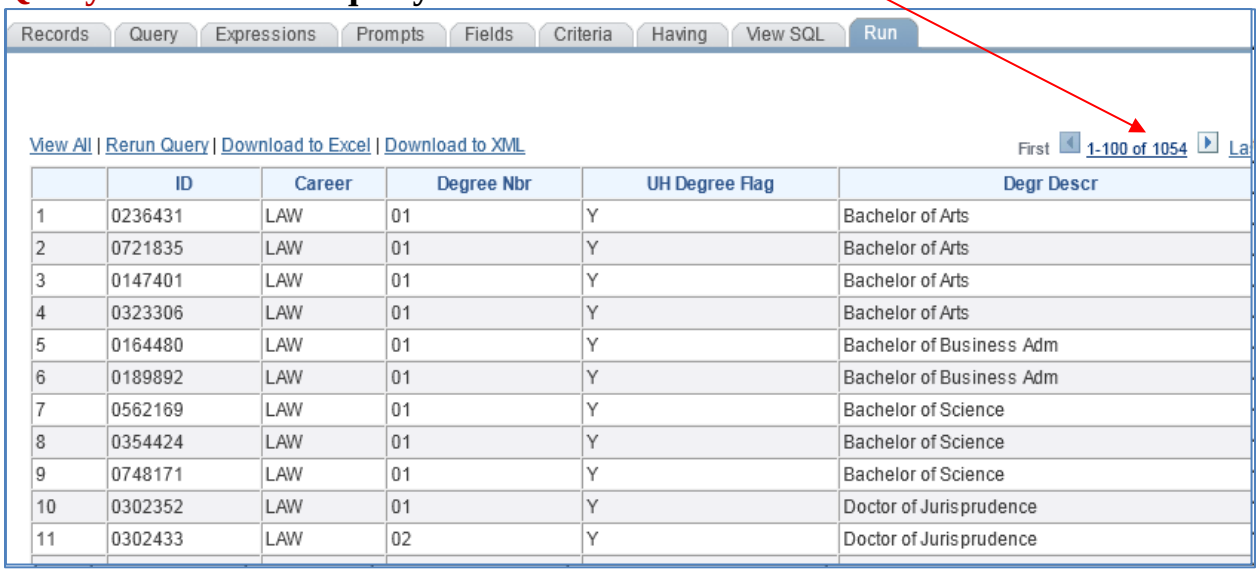


Step 11: Repeat these steps to add the outer join to B.EMPLID.



Step 12: Set the fields data sort to order by Degree Description and then rerun the query.

Query Results – The query now returns 1054 records.



	ID	Career	Degree Nbr	UH Degree Flag	Degr Descr
1	0236431	LAW	01	Y	Bachelor of Arts
2	0721835	LAW	01	Y	Bachelor of Arts
3	0147401	LAW	01	Y	Bachelor of Arts
4	0323306	LAW	01	Y	Bachelor of Arts
5	0164480	LAW	01	Y	Bachelor of Business Adm
6	0189892	LAW	01	Y	Bachelor of Business Adm
7	0562169	LAW	01	Y	Bachelor of Science
8	0354424	LAW	01	Y	Bachelor of Science
9	0748171	LAW	01	Y	Bachelor of Science
10	0302352	LAW	01	Y	Doctor of Jurisprudence
11	0302433	LAW	02	Y	Doctor of Jurisprudence

Step 13: Save the query.

Chapter 11: In-Class Exercise (Multiple Table Joins)

Query 1

Develop a query that displays the EMPLID and CUM_GPA for all students enrolled as of census during the Fall of 2007 at UH-Main if the student had a cumulative grade point average greater than 2.5. List the UHIR_DEGR_DESCR and UHIR_DEGR_DT as well, if the student obtained a UH degree in 2005 – 2007. Save the query as UHIR_TRAIN_JOIN2.

ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.

CHAPTER 12 – Creating Queries that Use Subqueries

Overview A subquery, sometimes called a sub-SELECT, is a query whose results are used by another query. The main query uses the subquery's result set as a comparison value for a selection criterion. You create a subquery when you need to compare a field value to the results of a second query.

Example *The UHIR_TRAIN_JOIN query (from Chapter 11) will display the LAW students who were enrolled in fall of 2007. A subquery will be added to identify which of those students were enrolled in spring of 2008 as well.*

Subquery Basics

Subqueries are also referred to as *sub-SELECT* queries and *Nested Queries*. Subqueries function to pass values up to a higher query in filtering out data that should be extracted.

Subqueries are also needed in using the PeopleSoft Query Tool when joined tables involve a record that has Effective Dated fields that are key fields and it is desired to use an Outer Join connection between the tables. SQL function NVL is very useful when designing queries having this scenario. For additional information on using the Null Value function, click on this link to refer to Appendix section NVL.

Subquery Rules

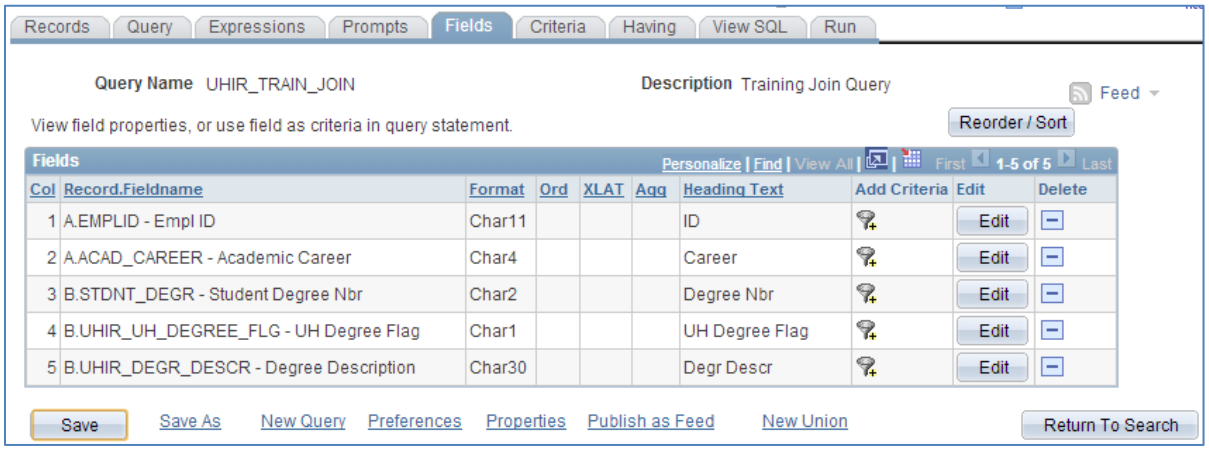
There is really only one rule that distinguishes subqueries, the one rule is:

1. **Return Only a Single Field/Expression**. A subquery can have only one field selected as output that is returned to the Main Query (Top Level Query). However multiple subqueries can be used in a main query.

Creating a Subquery Query

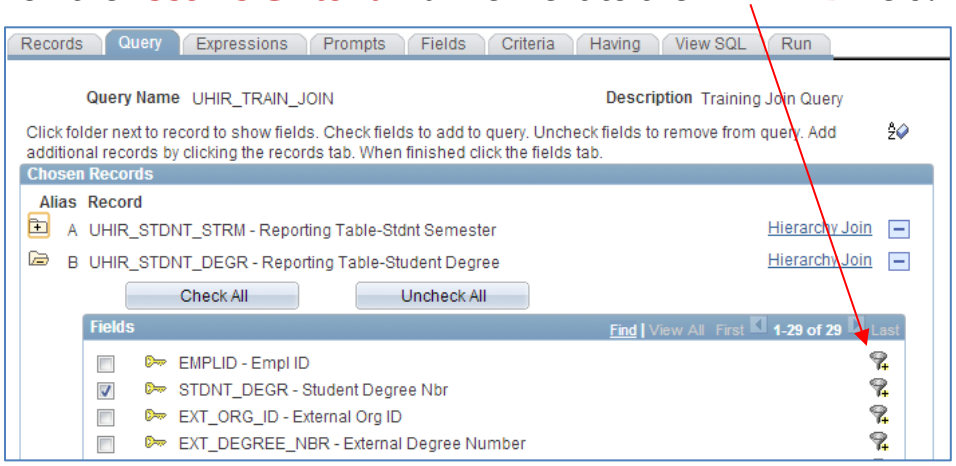
The below steps illustrate the creation of a query with a subquery:

Step 1: Pull-up the UHIR_TRAIN_JOIN query (if not already open).



Step 2: Click on the 'Query' tab.

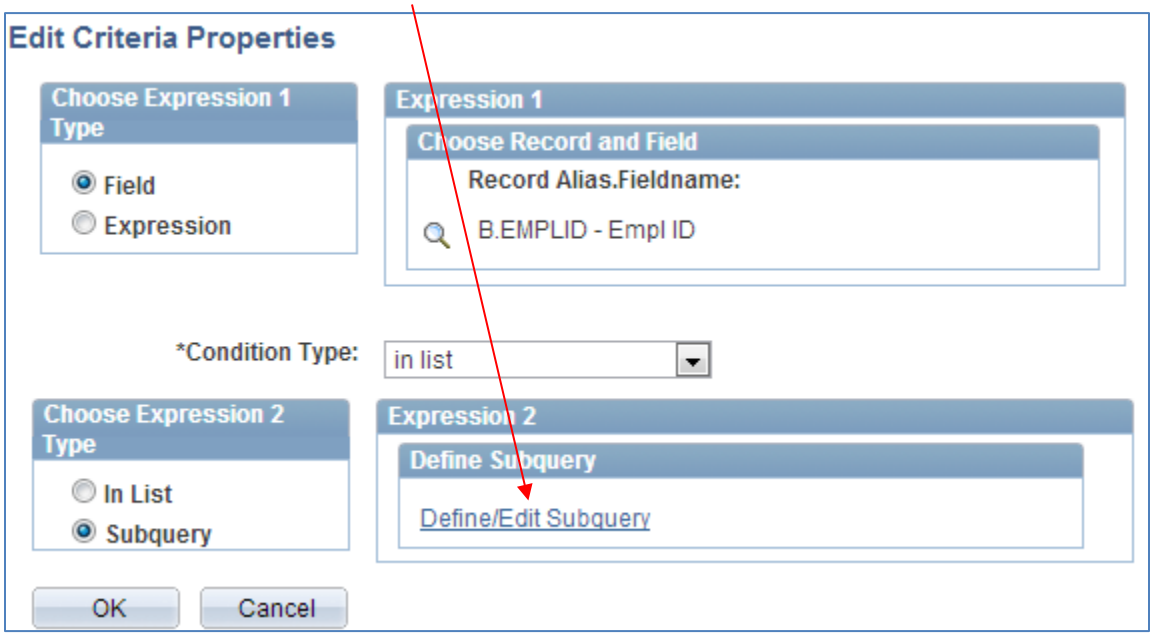
Step 3: Click on the 'Use As Criteria' funnel next to the EMPLID field.



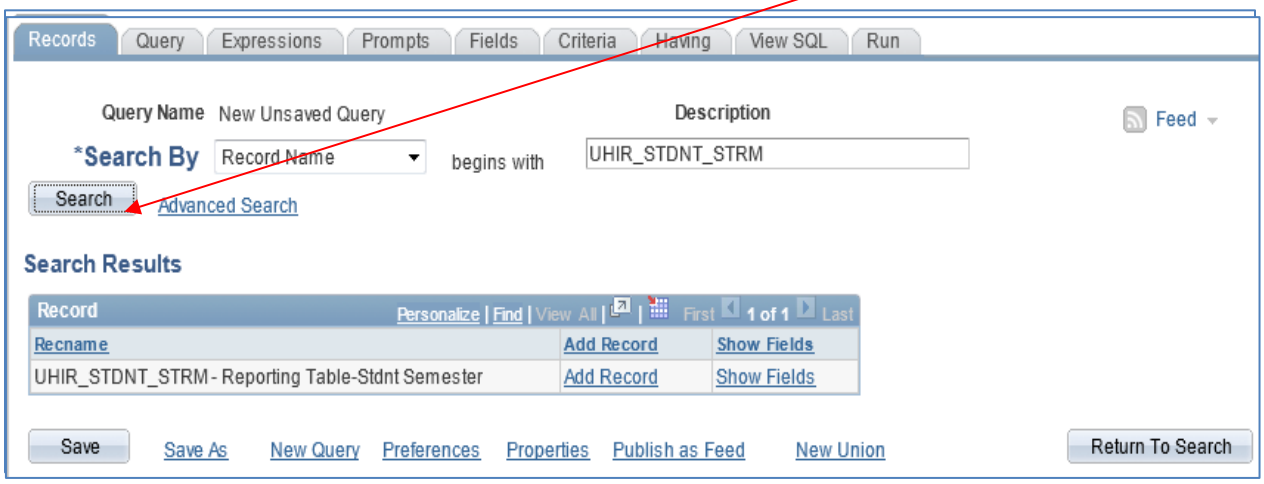
Step 4: Enter 'In List' condition type.

Step 5: Click on the 'Subquery' radio button

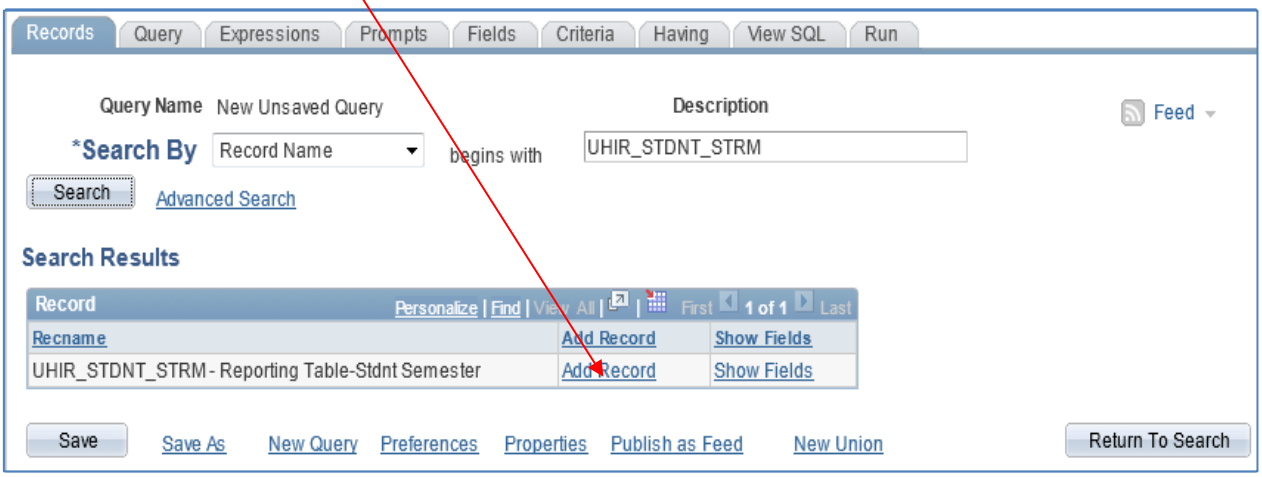
Step 6: Click on Define/Edit Subquery link.



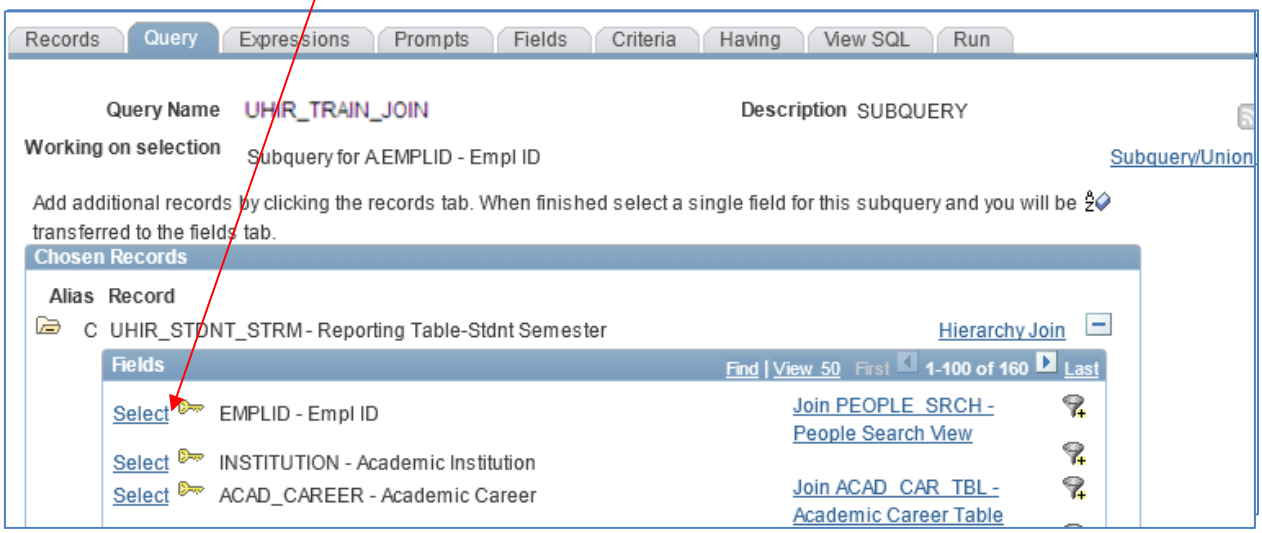
Step 7: Type UHIR_STDNT_STRM and click on the 'Search' button.

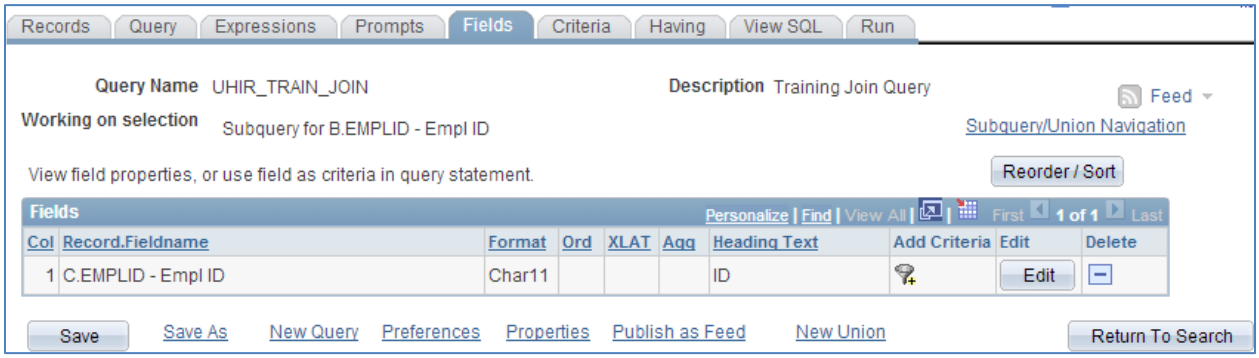


Step 8: Click the Add Record link.



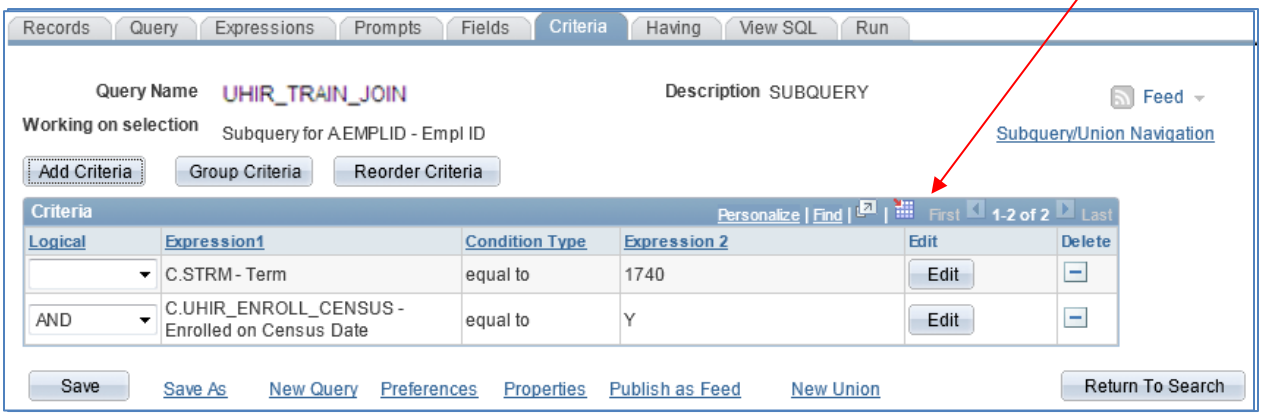
Step 9: Click the Select link next to the EMPLID field.





Step10: Click on the Criteria tab.

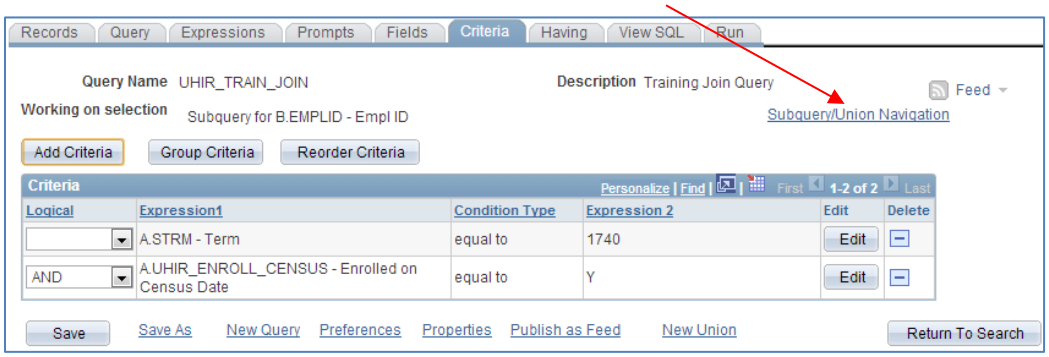
Step 11: Enter Criteria arguments until the Criteria page looks like this.



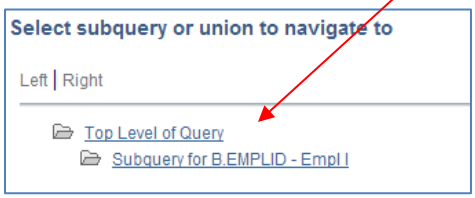
TIP:

TIP: Subquery Navigation

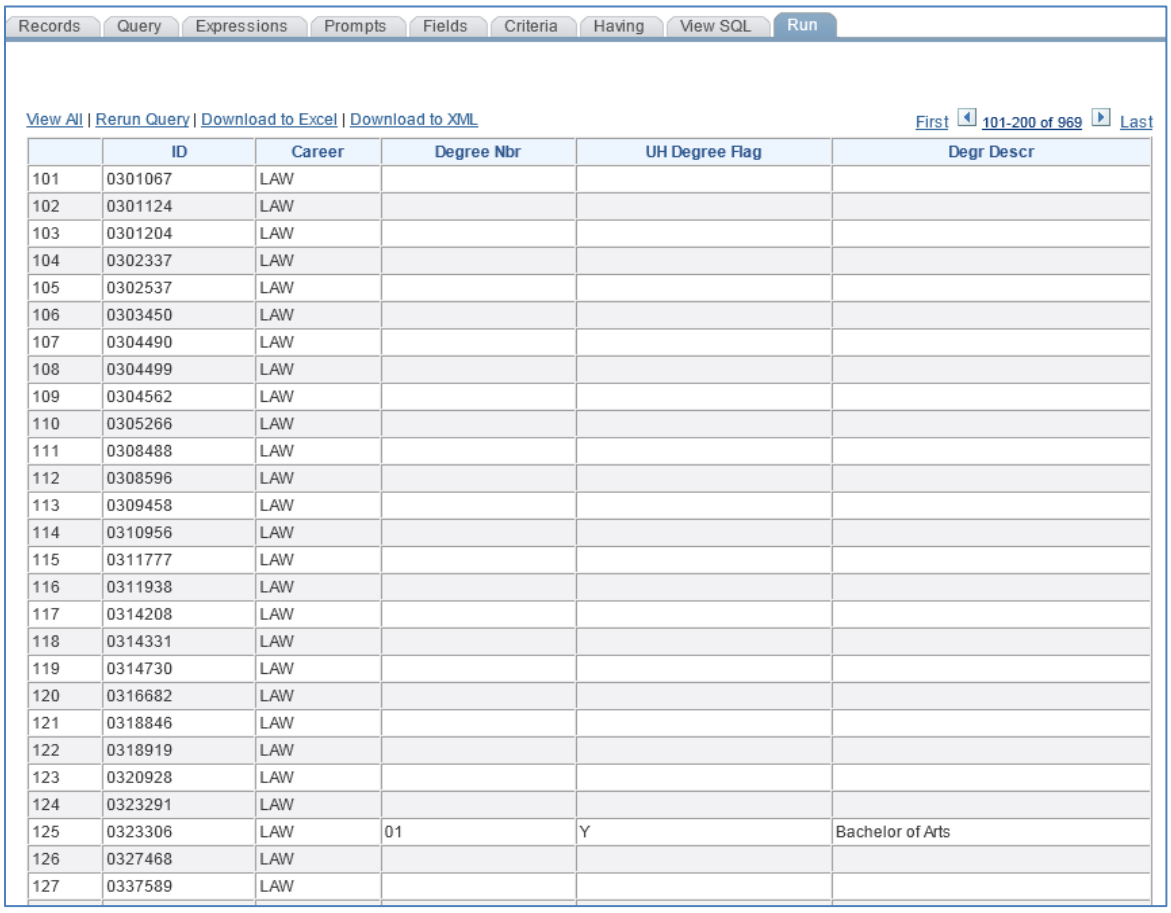
Step 12: Click on the Subquery/Union Navigation link.



Step 13: Click the Top Level Query link.



Step 14: Click on the 'Run' tab to run and view the output and Save as UHIR_TRAIN_SUBQUERY.



	ID	Career	Degree Nbr	UH Degree Flag	Degr Descr
101	0301067	LAW			
102	0301124	LAW			
103	0301204	LAW			
104	0302337	LAW			
105	0302537	LAW			
106	0303450	LAW			
107	0304490	LAW			
108	0304499	LAW			
109	0304562	LAW			
110	0305266	LAW			
111	0308488	LAW			
112	0308596	LAW			
113	0309458	LAW			
114	0310956	LAW			
115	0311777	LAW			
116	0311938	LAW			
117	0314208	LAW			
118	0314331	LAW			
119	0314730	LAW			
120	0316682	LAW			
121	0318846	LAW			
122	0318919	LAW			
123	0320928	LAW			
124	0323291	LAW			
125	0323306	LAW	01	Y	Bachelor of Arts
126	0327468	LAW			
127	0337589	LAW			

Chapter 12: In-Class Exercise (Subqueries)

Query 1

Your saved 'UHIR_TRAIN_JOIN' query will display the UHM Law students who were enrolled as of census Fall 2007 with degree information for the student if the student's degree was received in 2007 and was a UH degree. For this population of students, add a subquery to identify which of those students were also (a) enrolled at the end of Fall 2008 and (b) during Fall 2008 was assigned to activity group (UHIR_ACTIVITY_TYPE) 'R3' which is the UH Communication Group. Save the query as UHIR_TRAIN_SUBQUERY2.

ENTER YOUR PEOPLESOFT ID AND NAME FOR THE DESCRIPTION WHEN SAVING THE QUERY.

CHAPTER 13 – Creating Unions

Overview

Unions enable you to get the results from two or more separate queries at the same time. You can create a union of multiple queries only when the queries have the following common elements:

- The same number of selected fields.
- The same data types for all fields.
- The same display order for the columns.

Example

The query developed in this chapter uses a union to list the students that meet the following criteria: The student must have been enrolled at UH-Main during the 1740 semester. In addition, the student must have an SAT test score of 1200 and/or a GPA of at least 3.5. The first query will return those students that have CUM_GPA of 3.5 or higher. The second query in the union will return those students that scored a 1200 or higher on their SAT.

Union Queries Basics

Union queries are useful when you want to combine the results from two or more separate queries into a single output that is run at the same time.

UNION (Distinct) versus UNION ALL

The PeopleSoft Query Tool **automatically creates union queries as UNION (Distinct)** versus UNION ALL. With the UNION command only distinct values are selected. However with the UNION ALL command, all values are selected including duplicate occurrences. A UNION statement effectively does a SELECT DISTINCT on the results set.

For Example:
Table 1 Data: First, Second, Third, Fourth, Fifth
Table 2 Data: First, Second, Fifth, Sixth
Result Set with UNION: First, Second, Third, Fourth, Fifth, Sixth *(This will remove duplicate values)*
Result Set with UNION ALL: First, First, Second, Second, Third, Fourth, Fifth, Fifth *(This will repeat values)*

Union Query Rules

There are three rules that must be followed when creating union queries. The rule are:

1. **Same Number of Fields**. You must have the same number of selected fields in the independent queries.
2. **Same Field Data Types**. You must use the same data type for corresponding fields.
3. **Same Column Ordering**. You must display corresponding fields in the same order.

Creating a Union Query

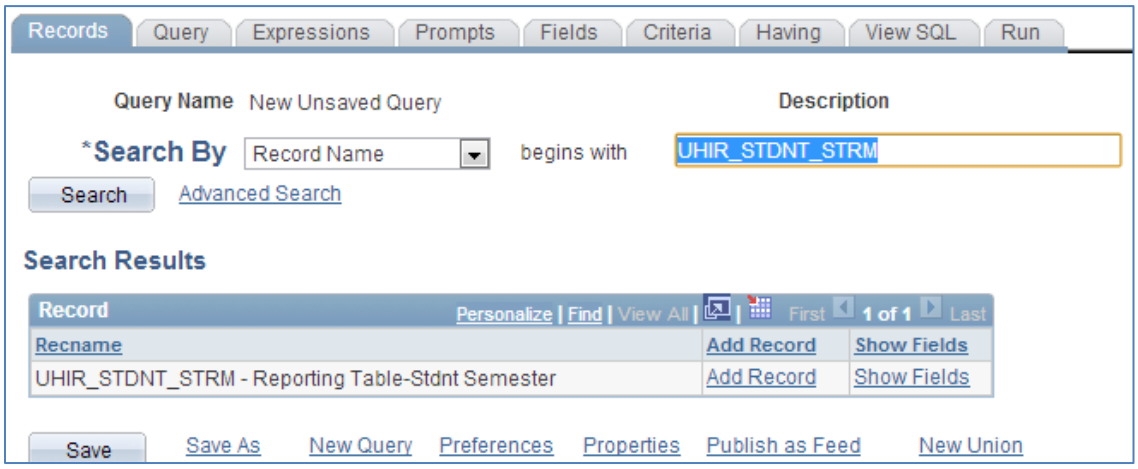
The below steps illustrate the creation of a union query:

Step 1: Navigate to Query Manager (if it is not already open).

Step 2: Click on the Create New Query tab.

Step 3: Type UHIR_STDNT_STRM in the 'begins with' edit box and click Search **.**

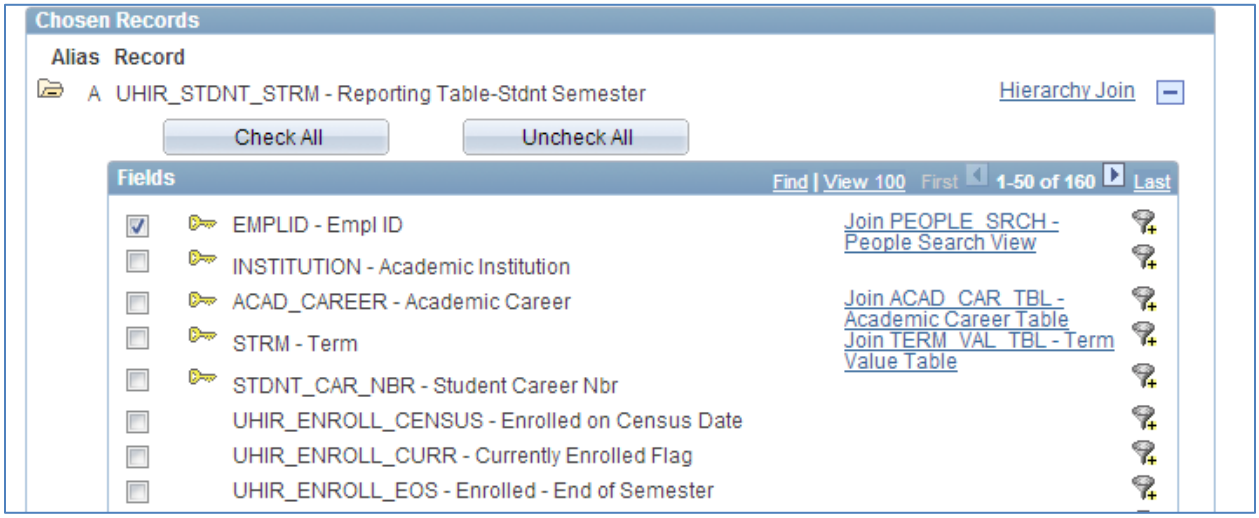
Step 4: Click the Add Record link.



Step 5: Select the criteria to limit the result set to:

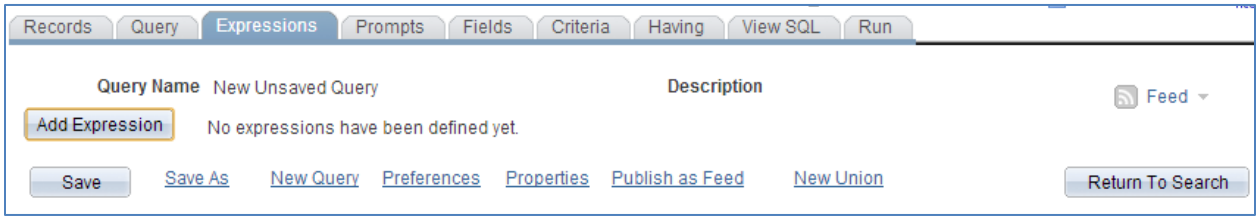
1. **Institution '00730'**
2. **STRM '1740'**
3. **CUM_GPA not less than '3.5'**

Step 6: Select the EMPLID and CUM_GPA fields so that they are displayed on the report:



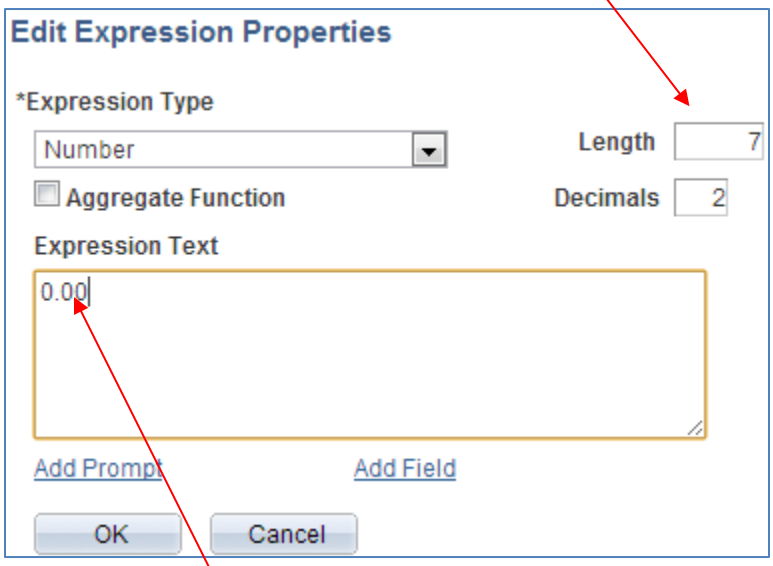
Step 7: Now you need to add a dummy field that matches the UHIR_SAT_COMP test score data type. Although this query will not return the actual SAT test score values, it is necessary to add the field so that the union query (to be added in steps 14 – 24) matches in terms of number of columns and data types for each column.

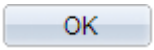
To add this dummy SAT Score Field, click on the Expressions tab and then on the Add Expression **Button.**



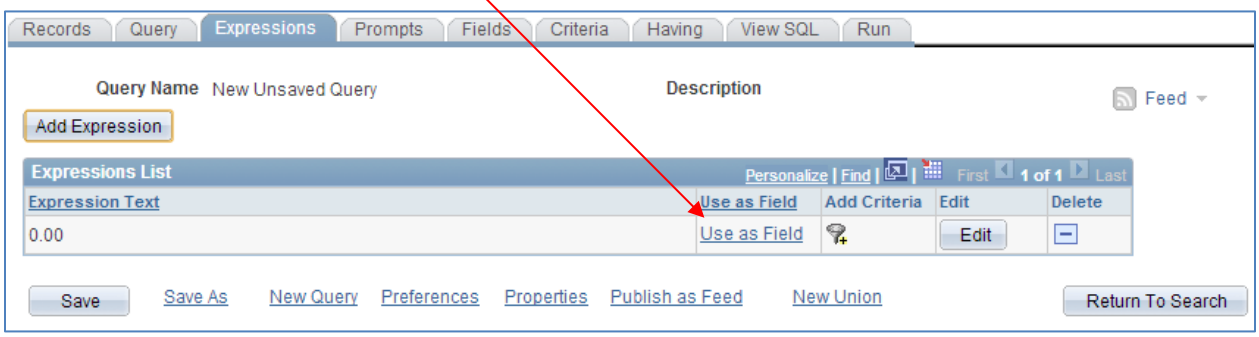
Step 8: In the Edit Expression Properties select the Expression Type (Number), and enter the Length (7) and Decimals (2).

Note: Remember that this dummy field must match the UHIR_SAT_COMP field on the second query in the union that will be created in steps 14 – 24.

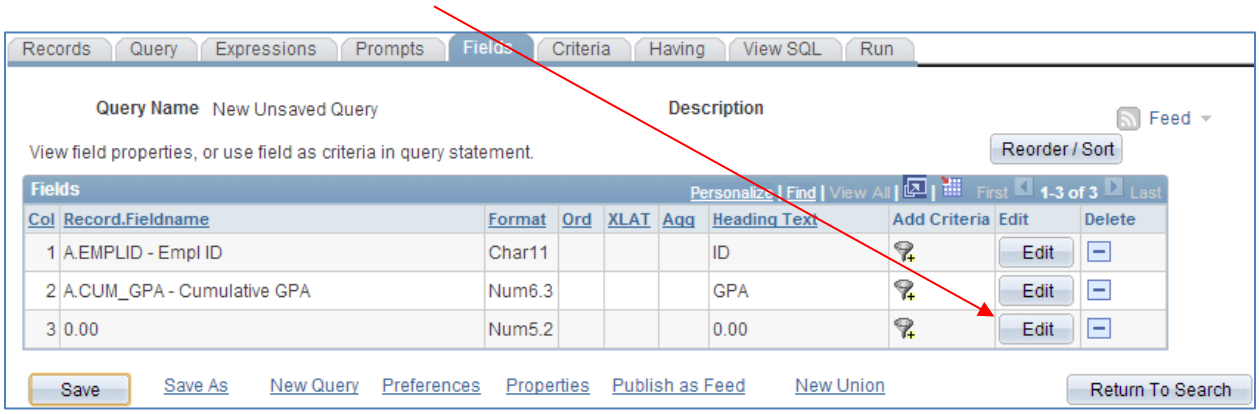


Step 9: Type 0.00 in the 'Expression Text' box and Click  **.**

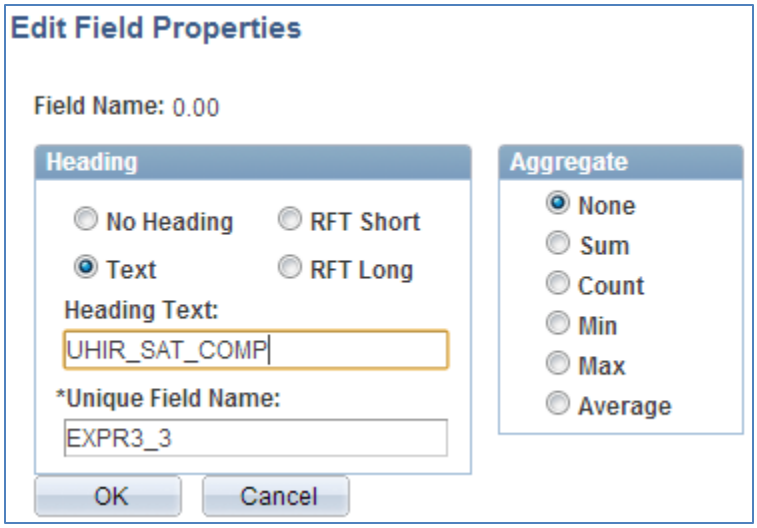
Step 10: Click on the Use as Field link.



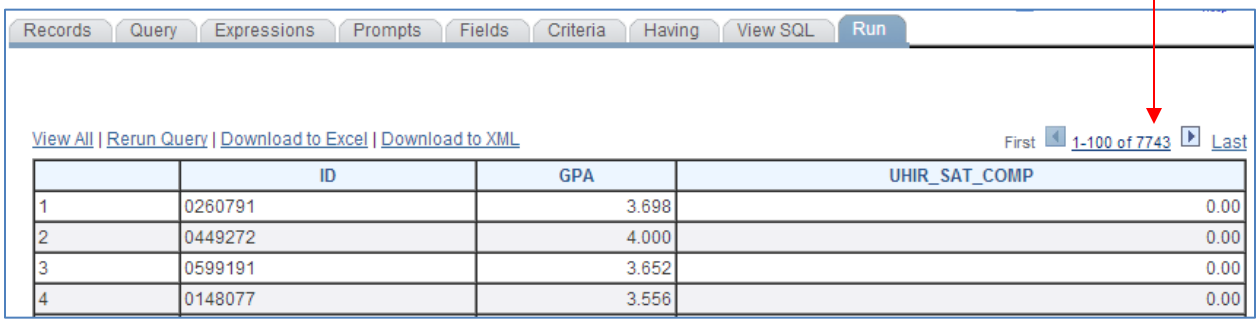
Step 11: Click on the Edit next to the dummy 0.00 field to change the heading.



Step 12: Type UHIR_SAT_COMP into the 'Heading Text' edit box and click OK.

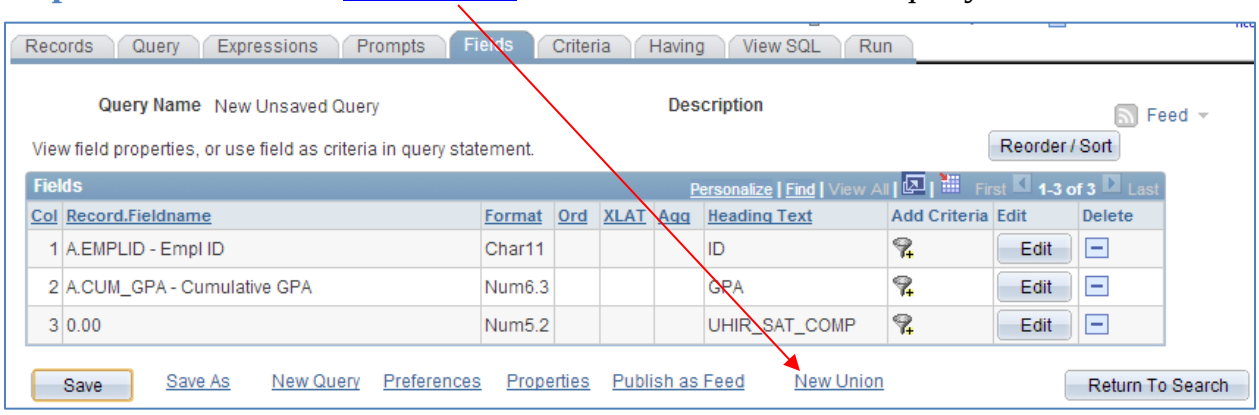


Step 13: You can now Run the query. Make a note of the number of records returned.



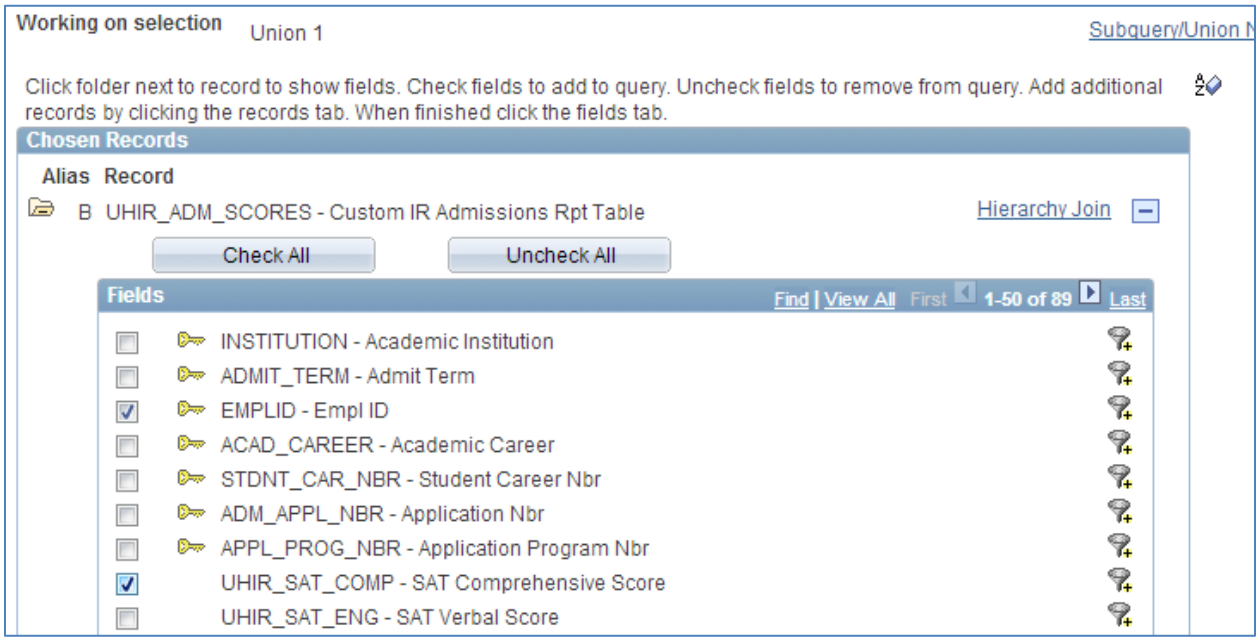
Note: At this point you are done creating the first of two queries that will be used in the **union** together. Steps 14 – 24 will step you through creating the second query in the union.

Step 14: Click on the New Union link to create the second query of the union**.**



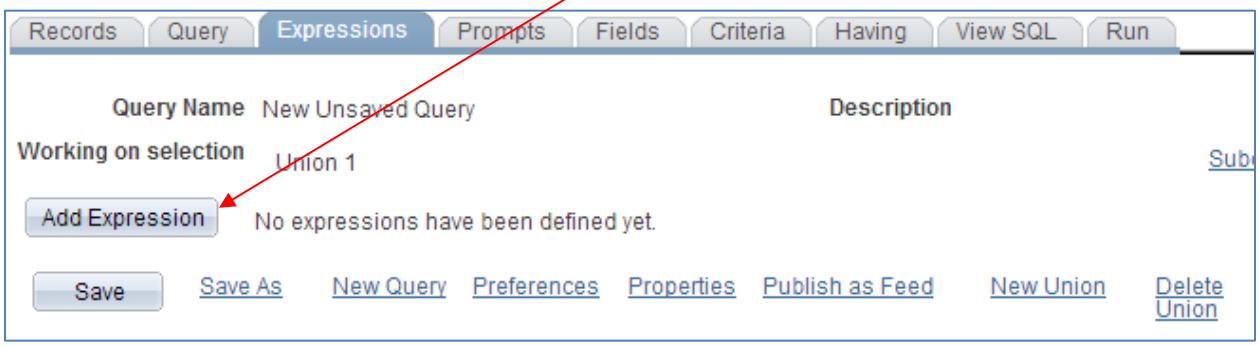
Step 15: Type 'UHIR_ADM_SCORES' in the begins 'with' edit box and click Search **. Then click** Add Record.

Step 16: Click on the 'EMPLID' and 'UHIR_SAT_COMP' checkboxes to display the fields on the report.



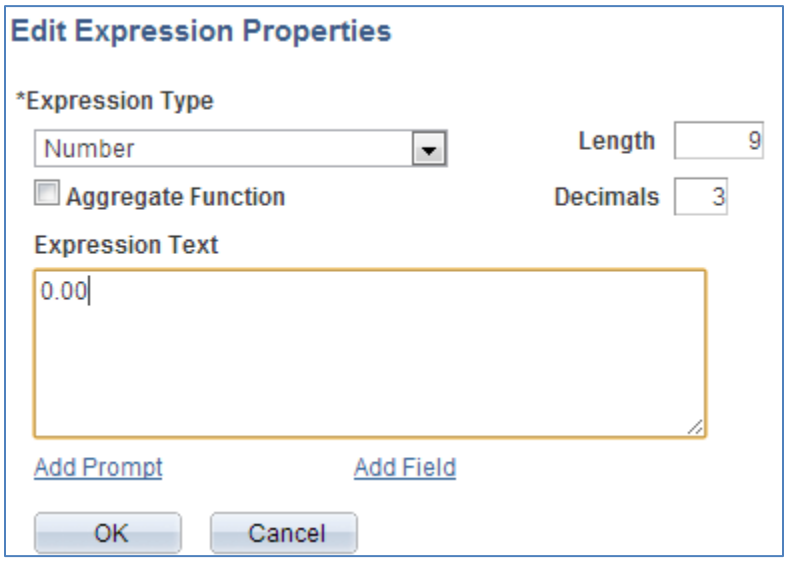
Step 17: Click on the Expressions tab and then on the Add Expression Button to add a dummy CUM_GPA field.

Note: Remember that the number of columns and the data types for the first and second query must match in order for a union to return results. This is why we are adding the dummy CUM_GPA field.

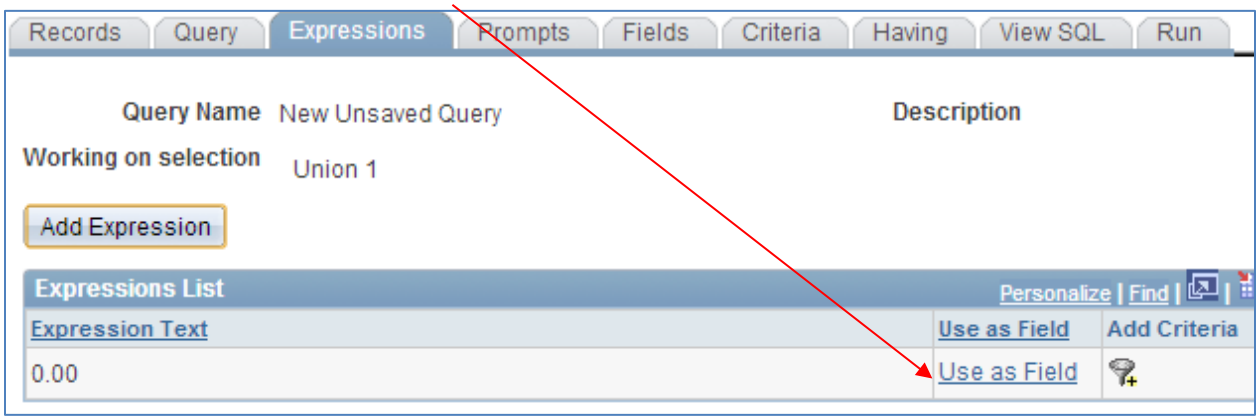


Step 18: Select the Expression Type (Number), and type the Length (9), and number Decimals (3), and Expression Text (0.00) and click OK.

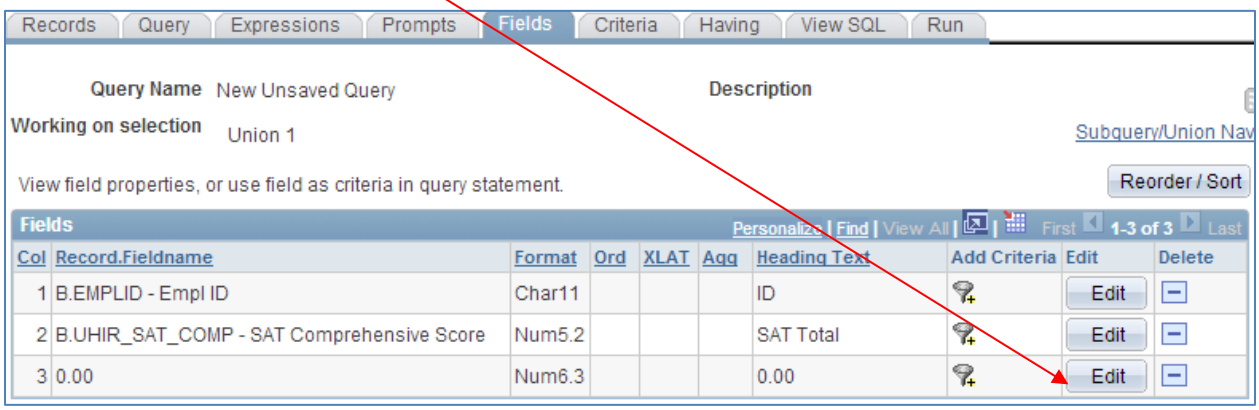
Note: Remember that these values **must match** the CUM_GPA field from the first query in the union that actually selects the CUM_GPA.



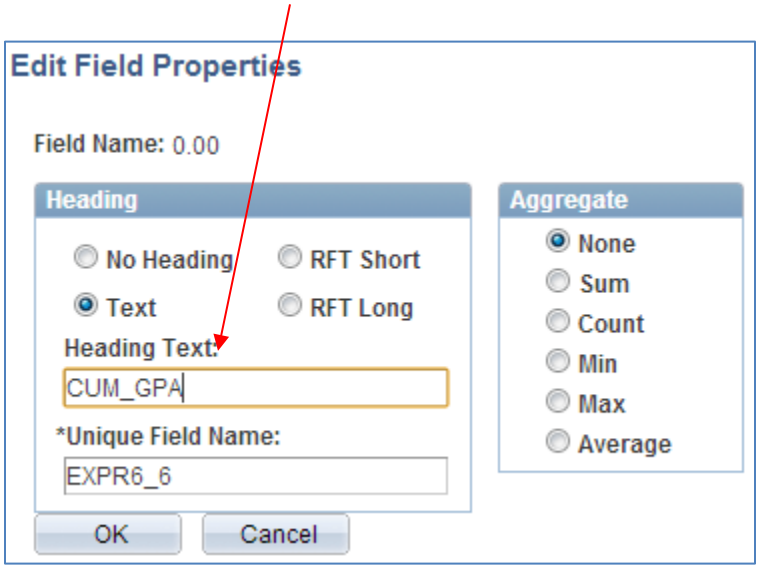
Step 19: Click on the Use as Field link.



Step 20: Click on the Edit next to the 0.00 column to change the heading.

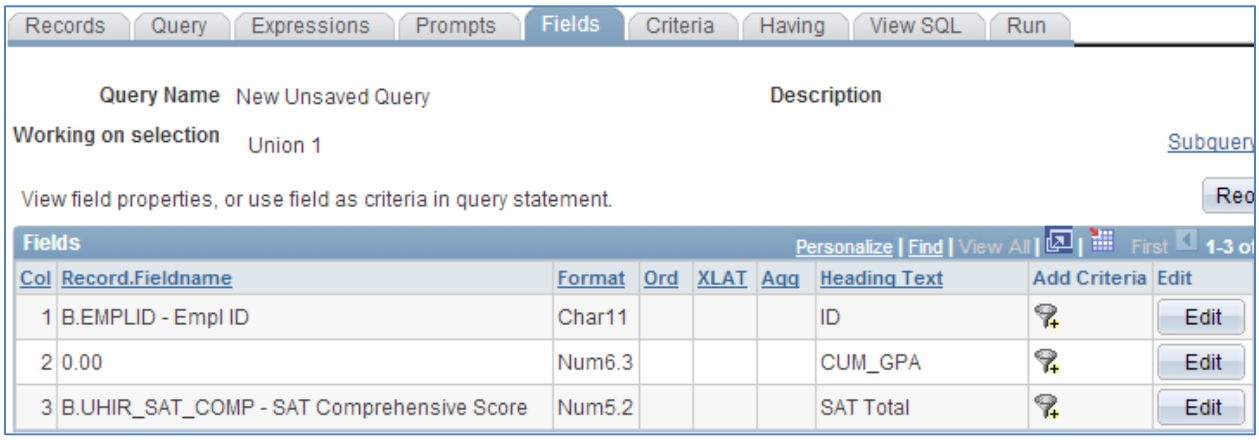


Step 21: Type CUM_GPA in Heading Text and click OK.

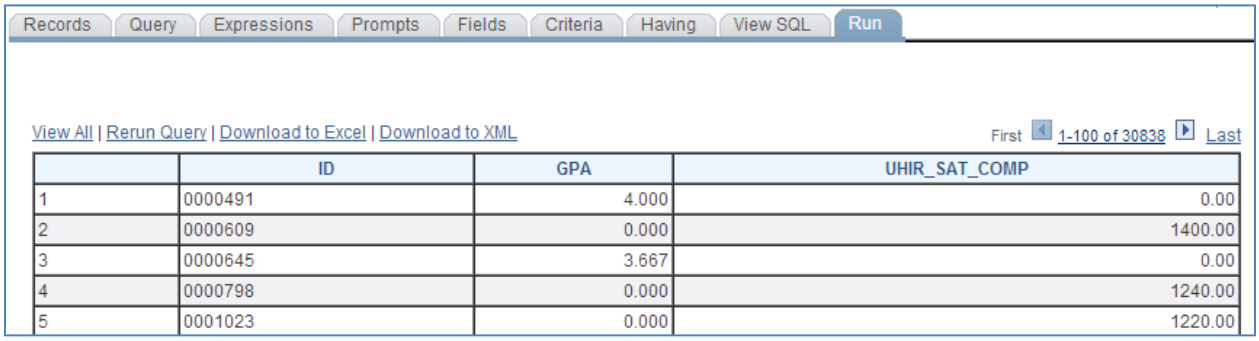


Step 22: Add the criterion to limit the students to those that scored a minimum of 1200 on their UHIR_SAT_COMP test.

Step 23: Change the order of the columns to match the first query.
1. EMPLID, 2. CUM_GPA, 3. UHIR_SAT_COMP



Step 24: Run the query to verify the results. You will notice that the number of rows is now greater than the number of rows in the initial query because students that scored 1200 or more on their SAT have been added the result set.

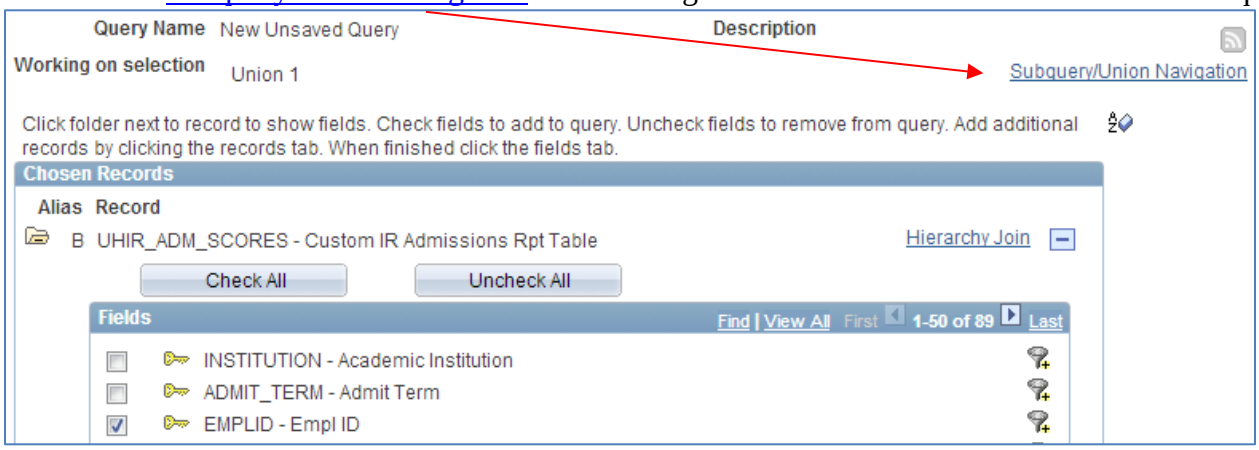


	ID	GPA	UHIR_SAT_COMP
1	0000491	4.000	0.00
2	0000609	0.000	1400.00
3	0000645	3.667	0.00
4	0000798	0.000	1240.00
5	0001023	0.000	1220.00

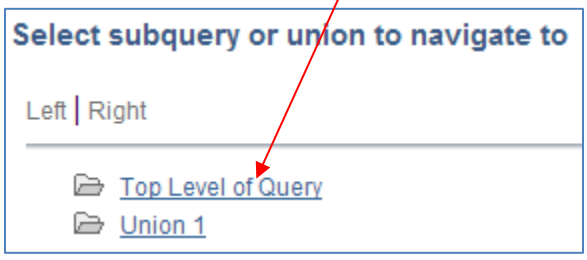
TIP:

TIP: Union Query Navigation

Click on the Subquery/Union Navigation link to navigate back and forth between the two union queries.



Click on the Top Level of Query link to go to the first query you created and the Union 1 to go to the second query you created.



Sample SQL Expression Functions

There are many more SQL/PeopleSoft functions and formats beyond the samples listed in this Appendix. Users are encouraged to continue to broaden and explore additional reference materials on SQL writing and functions. There are web search engines such as Google and Bing that provide SQL resources and websites such as The Oracle Database SQL Reference: http://www.oracle.com/pls/db10g/portal.portal_demo3?selected=1 for example.

Functions are special types of command words in the SQL command set, usually one-word commands which manipulate data items and return a single value which can be determined by various input parameters.

Function Statements are predefined system commands expressed with their operational parameters.

String Functions

String (also referred to as Character) functions perform operations on a string (char/varchar) input value and return a string or numeric value. Here are some functions that can be used on strings:

CONCAT(str1, str2) ; str1 || str2

The CONCAT function concatenates str1 and str2, returning str1 followed immediately by str2 (appends two or more literal expressions, column values or variables together into one string).

Alternatively, the bar or pipe (||) syntax can be used instead of the 'CONCAT' syntax to merge text strings together. "***CONCAT (str1, str2)***" is equivalent to "***str1 || str2***".

Example(s): If First Name is "Mickey" and Last Name is "Mouse"

- CONCAT(First Name, Last Name) returns 'MickeyMouse'
- First Name || Last Name returns 'MickeyMouse'
- CONCAT(First Name, ' ', Last Name) returns 'Mickey Mouse'
- First Name || ' ' || Last Name returns 'Mickey Mouse'

INITCAP(str1)

INITCAP converts a string to initial capital letters.

Example(s):

- INITCAP('mickey mouse') = Mickey Mouse
- INITCAP('MINNIE MOUSE') = Minnie Mouse

INSTR(str1,str2,[start position],[nth_appearance])

INSTR returns the location of a substring in a string. Str1 is the string to search and str2 is the substring to search for in str1. Start position is the position in str1 where the search begins (this argument is optional, if omitted the default is 1 which is the first position in the string). Nth appearance is the nth appearance of str2 (this argument is optional, if omitted the default is 1 which is the first position in the string).

Example(s):

- INSTR('Mickey','c') = 3 (first occurrence of the letter c)
- INSTR('Mickey Mousey','y',1,2) = 13 (second occurrence of the letter Y)

LENGTH(str1)

LENGTH returns the length of str1 in number of characters.

Example(s):

- LENGTH('ABC DEF') returns 7 (the space is included)

LOWER(str1)

LOWER returns a copy of str1 with all uppercase letters changed to lowercase. This function is similar to initcap but focuses on the entire string

Example(s):

- LOWER('ABCdeF') returns abcdef

REPLACE(str1,str_to_replace,[replacement_str])

REPLACE replaces a sequence of characters in a string with another set of characters. Str1 is the string being affected. Str_to_replace is the string which will be searched for in str1 and overwritten, this argument is optional (if omitted, the replace function removes all occurrences of str_to_replace and returns the resulting string). Replacement_str is the string that will overwrite the occurrences of str_to_replace.

Example(s):

- REPLACE('Mickey the Rat','Rat','Mouse') = Mickey the Mouse

SUBSTR(str1, start[, how_many])

SUBSTR returns the substring – a part of a string – of str1, starting at position start (extracts a portion of a string or field). If how_many is not provided, all characters from start through the end of the string are returned. If how_many is provided, the substring beginning at position start and having a length of how_many are returned. If position is 0, then it is treated as 1. If position is positive, then the count starts from the beginning. If position is negative, then it starts from the end and counts backward. How_many is the number of characters to extract.

Example(s):

- SUBSTR('ABCDEFG',1,3) returns 'ABC'
- SUBSTR('ABCDEFG',3,4) returns 'CDEF'
- SUBSTR('ABCDEFG',4) returns 'DEFG'
- SUBSTR('ABCDEFG',-5,3) returns 'CDE'

UPPER(str1)

UPPER returns a copy of str1 with all lowercase letters changed to uppercase.

Example(s):

- UPPER('Abcdef') returns ABC DEF

Aggregate Functions

Aggregate functions operate against a collection of values, but return a single, summarizing value. The PeopleSoft Query Tool has built-in AVG, SUM, COUNT, MIN and MAX aggregate functions. These aggregates can be applied directly to a selected field of output by using the 'Edit Field Properties' subpage of the 'Fields' tab . An aggregate function is a special type of operator that returns a single value based on multiple rows of data. When your query includes one or more aggregate functions, PeopleSoft Query collects related rows and displays a single row that summarizes their contents.

For additional information on the Aggregate Functions, go to this section of the manual: Aggregates .

Numeric Functions

Numeric functions perform operations on numeric values and returns numeric values. Here are some functions that can be used on numeric values:

ABS(n)

ABS returns the absolute value of n. If n is positive or zero, the result is n. If n is negative, the result is n without the negative sign.

Example(s):

- ABS(-5) returns 5.

CEIL(n)

CEIL returns the ceiling of n, which is the smallest integer equal to or greater than n (returns the smallest integer value that is greater than or equal to a number). For example, CEIL(5.2) is 6, since 5 is not equal to or greater than 5.2, but 6 is greater than 5.2. Note that
CEIL(-4.4) is -4. CEIL is equivalent to rounding up.

FLOOR(n)

FLOOR returns the floor of n, which is the largest integer equal to or less than n (returns the largest integer value that is equal to or less than a number). For example, FLOOR(8.7) is 8, since 9 is not equal to or less than 8.7, but 8 is less than 8.7. Note that
FLOOR(-3.2) is -4. FLOOR is equivalent to rounding down.

ROUND(n[, dec])

ROUND returns n rounded to dec decimal places, or to the nearest integer if dec is not provided (returns a number rounded to a certain number of decimal points). If dec is negative, the value is rounded to dec powers of 10 (-1 to the nearest 10, -2 to the nearest 100, etc.).

Example(s):

- ROUND(12.3) returns 12
- ROUND(12.8) returns 13
- ROUND(-12.3) returns -12
- ROUND(-12.8) returns -13
- ROUND(12345.6789, 2) returns 12345.68
- ROUND(12345.6789, 1) returns 12345.7
- ROUND(12345.6789, 0) returns 12346

- ROUND(12345.6789, -1) returns 12350
- ROUND(12345.6789, -2) returns 12300

SQRT(n)

SQRT returns the square root of n. The square root is the number, which multiplied by itself, results in n.

TRUNC(n[, dec])

TRUNC returns n truncated to dec decimal places (returns a number truncated to a certain number of decimal points). If dec is zero or not provided, all digits after the decimal are dropped. If dec is negative, ABS(dec) digits to the left of the decimal are replaced by zero, and all digits after the decimal are dropped.

Example(s):

- TRUNC(12.3) returns 12
- TRUNC(12.8) returns 12
- TRUNC(-12.3) returns -12
- TRUNC(-12.8) returns -12
- TRUNC(12345.6789, 2) returns 12345.67
- TRUNC(12345.6789, 1) returns 12345.6
- TRUNC(12345.6789, 0) returns 12345
- TRUNC(12345.6789, -1) returns 12340
- TRUNC(12345.6789, -2) returns 12300

MOD(m,n)

MOD returns the remainder of m divided by n (and returns m if n is 0). It utilizes FLOOR functionality.

REMAINDER(m,n)

REMAINDER returns the remainder of m divided by n. It utilizes ROUND functionality

Example(s):

- REMAINDER(16,3) = 1
- REMAINDER(16,6) = 4
- REMAINDER(16,0) = 16
- REMAINDER(-16,3) = -1

ROWNUM

ROWNUM assigns a number indicating the order in which each row is returned by a query result set.

Example(s):

- ROWNUM = 1, 2 3…n for each row returned in the query results

Date Functions

Date and Time: Perform operations on a date and time input values and return string, numeric, or date and time values. Here are some functions that can be used on dates:

ADD_MONTHS(dt, m)

ADD_MONTHS returns the date that is m months in the future of dt (returns a date plus *m* months).

Example(s): Assume the current day is July 24, 2013

- ADD_MONTHS(SYSDATE,3) = 10/24/2013
- ADD_MONTHS(SYSDATE,-3) = 04/24/2013

SYSDATE

SYSDATE returns the current system date and time. Note that it does not take any arguments.

Example(s): Assume the current day is July 24, 2013

- TO_CHAR(SYSDATE-30, 'MM-DD-YY') = 06-24-13

MONTHS_BETWEEN(date1, date2)

MONTHS_BETWEEN returns number of months between two dates.

Example(s): Assume the current day is July 24, 2013 and Start Date is August 20 2007

- MONTHS_BETWEEN(SYSDATE,A.START_DT) = 71

NEXT_DAY(date, char)

NEXT_DAY returns the date of the first weekday named that is later than the date specified.

Example(s): Assume the Start Date is July 24, 2013

- NEXT_DAY(A.START_DT,'MONDAY') = 07/29/2013

TRUNC(datetime)

TRUNC removes the time component from a date-time value

Example(s): Assume the Start DateTime is July 24, 2013 10:20:19AM

- TRUNC(StartDateTime) = 07/24/2013

Conversion Functions

Conversion functions change or convert values from one data type to another (character to numeric, numeric to character, character to date, or date to character).

Note: There are two things you should notice regarding the differences between numeric data types and character string types:

1. Arithmetic expressions and functions can be used on numeric values.
2. Numeric values are right-justified, whereas character string data types are left-justified in the output result.

Here are some functions that can be used to convert values from one type (date, number, string) to another:

TO_CHAR(value[, fmt])

TO_CHAR converts a number, date, or part of a date into a string. Value is either a number or date that will be converted to a string. Fmt (format_mask) is the format used to convert the value to a string.

For date conversions, if fmt (format mask) is not provided, the date is returned in the default format for the system, usually 'DD-MON-YY'. (In this format, 2/13/2009 would be '02-FEB- 09'.)

For number conversions, if fmt (format mask) is not provided, a default format will be used. This function is typically used to format numbers with commas, periods, currency symbols, and leading zeroes.

The fmt argument is a string containing codes that instruct TO_CHAR how to format the value. There are many different format codes and several can be used at one time. Some of the format codes are:

- AM – the AM/PM indicator
- DD – day of the month
- HH – hour of the day (1-12)
- HH24 – hour of the day (0-23)
- MI – minute of the hour (0-59)
- MM – month (1-12, with 1 = January)
- MON – three-character abbreviation of the month
- SS – second of the minute (0-59)
- YY – two-digit year
- YYYY – four-digit year
- $ – display a dollar sign
- 0 – display a leading zero
- 9 – display a digit
- , – display a comma
- . – display a period
- FM – disable leading and trailing spaces

Punctuation such as dashes, colons, and slashes are included in the result string in the positions given in the fmt.

Example(s): TO_CHAR(date[,fmt])

Assume the current date is March 15, 2009 and the time is 4:25 PM. Here are several examples of how TO_CHAR would display this date:

- TO_CHAR(SYSDATE, 'MM/DD/YYYY') returns 03/15/2009
- TO_CHAR(SYSDATE, 'DD/MM/YYYY') returns 15/03/2009
- TO_CHAR(SYSDATE, 'YYYYMMDD') returns 20090315
- TO_CHAR(SYSDATE, 'HH:MI') returns 04:25
- TO_CHAR(SYSDATE, 'HH24:MI') returns 16:25
- TO_CHAR(SYSDATE, 'MM/DD/YYYY HH24:MI:SS') returns 03/15/2009 16:25:00
- TO_CHAR(SYSDATE, 'MM') returns 03 (this could be used for grouping in order to aggregate results by month)

Example(s): TO_CHAR(number[,fmt])

- TO_CHAR('1234567.89') returns 1234567.89
- TO_CHAR('1234567.89', '9,999,999') returns 1,234,568 (note the automatic rounding!)
  *there is a leading space in front of the number to leave room for a negative sign
- TO_CHAR('123', '099999') returns 000123
  *there is a leading space in front of the number to leave room for a negative sign
- TO_CHAR('123', 'FM099999') returns 000123; there is no leading space
- TO_CHAR('12345.60', '$999,999.99') returns $12,345.60
  *there are two leading spaces, one to leave room for the sign and one because there is no hundred-thousands digit

TO_DATE (string1,[format_mask])

TO_DATE converts a string to a date. String1 is the string that will be converted to a date. Format_mask is the format that will be used to convert string1 to a date.

Example(s): Assume the birthdate is March 11, 2007

- TO_DATE('March 11, 2007','MM/DD/YY') = 03/11/07
- TO_CHAR(TO_DATE(A.BIRTHDATE), 'MMDDYYYY') = 03112007 *(scenario where Expression Type is set to "Date")*
- TO_CHAR(TO_DATE(A.BIRTHDATE), 'MM/DD/YYYY') = 03/11/2007 *(scenario where Expression Type is set to "Character")*
- TO_CHAR(TO_DATE(A.BIRTHDATE), 'YYYY') = 2007 *(scenario where Expression Type is set to "Character", "Date", or "Number")*

TO_NUMBER (string,[format_mask])

TO_NUMBER converts a string to a number. String1 is the string that will be converted to a number. Format_mask is the format that will be used to convert string to a number, this argument is optional.

Example(s): Assume the A.EFFDT is July 24, 2013

- TO_NUMBER(TO_CHAR(TO_DATE(A.EFFDT),'YYYY')) = 2013

Condition Functions

These are functions that return different values based on a set of conditions that you specify.

DECODE(x, val1, [val2, result2, …, valN, result] [, default])

DECODE is used to produce a result that is based on the result of x. The x, val1, result1, etc. arguments can be numbers or strings. Up to 127 comparisons (value/result pairs) are allowed if there is not default value and up to 136 comparisons are allowed if there is a default.

The value of x is compared against val1, val2, etc. through valN, one at a time. When a match occurs, the corresponding result argument is returned. If x matches val1, DECODE returns result1, if x matches val2, DECODE returns result2, and so on. If there is no match, the default argument is returned, provided it is given; otherwise the special value NULL is returned.

Consider the following examples:

Example(s):

- DECODE(A.COLOR, 'R', 'Red', 'G', 'Green', 'B', 'Blue')
    - If A.COLOR is “R” then DECODE returns “Red”
    - If A.COLOR is “G” then DECODE returns “Green”
    - If A.COLOR is “B” then DECODE returns “Blue”
    - If A.COLOR is “Y” then DECODE returns NULL
- DECODE(A.DIRECTION, 0, 'North', 90, 'East', 180, 'South', 270, 'West', 'Unknown')
    - If A.DIRECTION is 0 then DECODE returns “North”
    - If A.DIRECTION is 90 then DECODE returns “East”
    - If A.DIRECTION is 180 then DECODE returns “South”
    - If A.DIRECTION is 270 then DECODE returns “West”
    - If A.DIRECTION is 360 then DECODE returns “Unknown”

You can also use the DECODE function to substitute a value when a field is blank. For date fields, an empty value is the special value NULL. To replace NULL values from the field A.DATE_FIELD with a default of date of 5/22/2009, use DECODE(A.DATE_FIELD, NULL, TO_DATE('05/22/2009', 'MM/DD/YYYY'), A.DATE_FIELD). Note the second A.DATE_FIELD as the last argument; this indicates that the original value of DATE_FIELD whenever DATE_FIELD is not NULL.

For text fields, an empty value is a single space. To replace a single space from the field A.TEXT_FIELD with the default string “N/A,” use DECODE(A.TEXT_FIELD, ' ', 'N/A', A.TEXT_FIELD). As with the previous example, the original value of the field is used when the field is not empty.

COALESCE(expr1, expr2, …, expr_n)

COALESCE returns the first non-null expression in the list (if all expressions evaluate to null, then the coalesce function will return null)

- IF-THEN’ functionality
- coalesce(mickey,minnie,goofy)
- IF mickey exists (not null) THEN result = mickey;
- ELSIF minnie exists (not null) THEN result = minnie;

- ELSIF goofy exists (not null) THEN result = goofy;
- ELSE result = null;
- END IF
- The coalesce function compares each value one by one

NVL(string1, replace_with) and NVL2(string1, value_if_not_null, value_if_null)

NVL allows substitution of a value when a null value is encountered. NVL2 allows the substitution of a value when a null value is encountered, as well as when a non-null value is encountered. NVL2 extends the functionality of NVL by letting you determine the value returned based on whether something is null or not null.

String1 is the string to be tested for a null value. Replace with is the value returned if string1 is null. Value_if_not_null is the value returned if string1 is not null. Value_if_null is the value returned if string1 is null.

Example(s):

- NVL(course_gpa,'Grade Pending')
  - if course_gpa is null then Grade Pending is returned otherwise course_gpa value is returned
- NVL2(FERPA,'Do Not Disclose','Disclose')
  - if FERPA is null then Disclose is returned otherwise DO Not Disclose is returned

CASE Expression

A **CASE** expression is a powerful tool for producing one of several possible results based on conditions. It is more flexible than DECODE, which just maps input values to output values. CASE expressions allow *ranges* to be mapped to output values and for multiple input fields to influence the output.

CASE expressions have the following format: CASE
WHEN condition1 THEN result1
[WHEN condition2 THEN result2]
…
[ELSE default_result] END

The CASE and END keywords are required. Each possible case is designated by the WHEN keyword, followed by a condition, the THEN keyword, and the result. The result must be a value or an expression that evaluates to a single value. The condition can be simple or complex, with many Boolean operators (such as AND and OR), as long as it evaluates to a result of true or false.

At least one WHEN clause must be given. All other WHEN clauses and the ELSE clause are optional. However, it is good practice to provide a default case with the ELSE clause in case unexpected values are encountered.

The conditions are examined in order from the first WHEN clause to the last WHEN clause. If condition1 is true, then the result of the CASE expression is set to result1. If condition1 is false, then condition2 is examined (if given); if condition2 is true, then the result of the CASE expression is set to result2. Each condition is examined until one that is true is encountered. If none of the conditions are true, the result of the CASE expression is set to default_result if an ELSE clause is provided; otherwise, the expression is set to NULL.

Consider this example of determining if a state is a Great Lakes state: CASE
WHEN A.STATE IN ('OH', 'MI', 'IN', 'IL', 'WI', 'MN', 'NY', 'PA') THEN 'Y'
ELSE 'N' END

If the STATE field contains one out of several state codes (OH, MI, etc.) then the CASE expression evaluates to “Y”. Otherwise, it evaluates to “N”.

Next, consider a version of this expression that takes into account that the Canadian province of Ontario is considered part of the Great Lakes region:

```
CASE
   WHEN A.COUNTRY = 'CAN' AND A.STATE = 'ON' THEN 'Y'
   WHEN A.COUNTRY = 'USA' AND A.STATE IN ('OH', 'MI', 'IN', 'IL', 'WI', 'MN', 'NY', 'PA')
THEN 'Y' ELSE 'N'
END
```

Each of the conditions includes an AND operator and examines data in two fields, COUNTRY and STATE. The conditions can include many conditions combined together.

The above could also be expressed as the following: CASE
WHEN
(A.COUNTRY = 'CAN' AND A.STATE = 'ON') OR
(A.COUNTRY = 'USA' AND A.STATE IN ('OH', 'MI', 'IN', 'IL', 'WI', 'MN', 'NY', 'PA')) THEN
'Y' ELSE 'N'
END

Here is an example with many conditions. This CASE expression translates a temperature in Fahrenheit into a description.

```
CASE
   WHEN A.TEMPERATURE < 30 THEN 'Very cold'
   WHEN A.TEMPERATURE BETWEEN 30 AND 44 THEN 'Cold' WHEN A.TEMPERATURE
BETWEEN 45 AND 59 THEN 'Cool' WHEN A.TEMPERATURE BETWEEN 60 AND 69 THEN 'Mild' WHEN
A.TEMPERATURE BETWEEN 70 AND 79 THEN 'Warm' WHEN A.TEMPERATURE BETWEEN 80 AND 89
THEN 'Hot'
   ELSE 'Very hot'
END
```

Assume that the value in A.TEMPERATURE is 53. The first condition, A.TEMPERATURE < 30, is examined and found to be false. The second condition, A.TEMPERATURE BETWEEN 30 AND 44, is examined and is also false. The third condition, A.TEMPERATURE BETWEEN 45 AND 59, is true. The result of the expression is “Cool”.

If the value of A.TEMPERATURE is above 89, none of the conditions in the WHEN clauses will be true, so the default value “Very hot” will be the result.

Note that since the order of the clauses is significant, the following CASE expression is equivalent:
CASE

```
   WHEN A.TEMPERATURE < 30 THEN 'Very cold' WHEN A.TEMPERATURE <= 44 THEN 'Cold' WHEN
A.TEMPERATURE <= 59 THEN 'Cool' WHEN A.TEMPERATURE <= 69 THEN 'Mild' WHEN
A.TEMPERATURE <= 79 THEN 'Warm' WHEN A.TEMPERATURE <= 89 THEN 'Hot'
   ELSE 'Very hot'
END
```

GREATEST(expr[,expr])

GREATEST returns the greatest occurrence from a list of one or more expressions.

Example(s): Assume the Test Date1 is July 24, 2013 and Test Date2 is July 1, 2013

- GREATEST(Test Date1, Test Date2) = 7/24/2013

LEAST(expr[,expr])

LEAST returns the least occurrence from a list of one or more expressions.

Example(s): Assume the Test Date1 is July 24, 2013 and Test Date2 is July 1, 2013

- LEAST(Test Date1, Test Date2) = 7/1/2013

Analytic Functions

Analytic functions compute an aggregate value based on a group of rows. They differ from aggregate functions in that they return multiple rows for each group. The group of rows is called a window. Analytic functions are the last set of operations performed in a query except for the final ORDER BY clause. All joins and all WHERE, GROUP BY, and HAVING clauses are completed before the analytic functions are processed.

Analytic functions are commonly used to compute cumulative, moving, centered, and reporting aggregates.

Calculations are independent of output.

Syntax: [.....] (.....) OVER (PARTITION BY .....)

- ..... (.....) OVER (PARTITION BY .....)
- ..... () OVER ()

- ..... (.....) OVER (PARTITION BY ..... ORDER BY ..... DESC)
  - ASC | DESC Specify the ordering sequence (ascending or descending)
  - ASC is the default

- ..... (.....) OVER (PARTITION BY ..... ORDER BY ..... DESC NULLS LAST)
  - NULLS LAST is the default for ascending order.
  - NULLS FIRST is the default for descending order.

Example(s):

- COUNT (A.EMPLID) OVER (PARTITION BY A.STRM)
- COUNT (A.EMPLID) OVER (PARTITION BY A.ACAD_LEVEL_BOT)
- COUNT (DISTINCT A.EMPLID) OVER (PARTITION BY A.ACAD_LEVEL_BOT)
- COUNT (DISTINCT A.EMPLID) OVER (PARTITION BY A.ACAD_LEVEL_BOT, A.STRM)
- SUM (A.UNT_TAKEN_PRGRSS) OVER (PARTITION BY A.EMPLID)
- ..... (.....) OVER (PARTITION BY A.ACAD_LEVEL_BOT ORDER BY A.CUM_GPA DESC)
- (PERCENT_RANK () OVER (PARTITION BY A.ACAD_LEVEL_BOT ORDER BY A.CUM_GPA DESC)) * 100
- PERCENT_RANK () OVER (PARTITION BY A.ACAD_LEVEL_BOT ORDER BY A.CUM_GPA DESC)
- ROW_NUMBER( ) OVER ([query_partition_clause] order_by_clause)

Troubleshooting Queries

Below are suggestions to follow on what to check when a query produces unexpected or incorrect results and/or errors:

If the query produces many more rows than are expected

- Are there any criteria missing?
  - Were you asked to limit the results to a particular term but there is no criteria on STRM?
  - Should you be showing only "active" rows, such as those with EFF_STATUS = 'A', PROG_STATUS = 'AC', etc.?
  - Are there criteria on the appropriate key fields, such as INSTITUTION = '00730' or '00759' or '00765' or ACAD_CAREER = 'UGRD' or 'GRAD', etc.?
- Are the joins between records correct?
  - If a join is missing, each row from the first table is combined with each row from the second table.
  - If you added the join yourself by creating criteria, did you join on all appropriate key fields? For example, if you are querying on SRVC_IND_DATA and join to SRVC_IND_RSN_TBL to get the description of the reason code, did you join on INSTITUTION and SRVC_IND_CD as well as SRVC_IND_REASON?
  - Did you use a left outer join instead of a standard join? Left outer joins are used only when you need rows from the first table to be shown regardless of whether there is a match in the second table. If a match is required, use a standard join.

If the query produces no results or many fewer rows than are expected

- Did you create a criterion on the wrong field? For example, to find a class by course ID, did you put a criterion on CATALOG_NBR instead of CRSE_ID?
- Did you compare against the wrong value? For instance, if you want active programs, did you check for PROG_STATUS = 'AC' or PROG_STATUS = 'ACTV'? (In this case, the former is correct; 'ACTV' is a valid value for PROG_ACTION.)
- Did you manually join two records on the wrong fields? If the values in each field are not like each other, there is unlikely to be a match, and if there is one, it will be by coincidence. For instance, the field ACAD_PLAN should not be joined to the field ACAD_PROG since they represent different things. (Hint: fields that can be joined will likely have similar names.)
- Is it impossible to meet the conditions? If you wrote a query to find cases where PROG_STATUS is either 'AC' or 'LA', did you use PROG_STATUS = 'AC' AND PROG_STATUS = 'LA' or PROG_STATUS = 'AC' OR PROG_STATUS = 'LA'? Since it is not possible for a field to have two values that the same time, a query with X = <value1> AND X = <value2> will always return no rows.
- Did you check that the tables you selected and fields you have added criteria to all have data in them. Isolate each table and appropriate field to confirm they have data.
- Check your Effective Dates.
- Check if there is an additional layer of Query Security being applied to one of the selected tables. Query Security is automatically applied in the background for certain tables where Query Security has been specified by the developer. You can identify tables having Query Security by reviewing the SQL tab.

Problems with expressions

- "Invalid identifier" error
  - Field names referenced in an expression must be spelled correctly (SRVC_IND_RSN is incorrect; SRVC_IND_REASON is correct)
  - Function names must also be spelled correctly (TRUNCATE is incorrect; TRUNC is correct)
  - Strings (literal text included in the expression) must be enclosed in single quotes
- Value displayed has too few characters or digits

- Check the Length property of the expression.
  - Character: If Length is 10 but the result of the expression is 20 characters long, only the first 10 characters are shown.
  - Number: Values are shortened by trimming digits off the end of the integer portion of the number (1234567.890 when displayed with a Length property of 5 is shown as 12345).

- Missing parentheses
  - Each opening parenthesis "(" must have a corresponding closing parenthesis ")"
  - Errors can vary depending on how the query engine interprets the generated SQL
- Functions
  - Did you include all required arguments? For example, SUBSTR takes two or three arguments, and the first two are required. Using SUBSTR('ABC') results in a "not enough arguments for function" error.
  - Did you include too many arguments? For example, LOWER takes only one argument. Using LOWER('ABC', 5) results in a "too many arguments for function" error.
  - Did you use the right types of arguments? MONTHS_BETWEEN takes two date arguments results in an "inconsistent datatypes" error. If a date is expected, a date must be given, if a number is expected, a number (or string that can be converted to a number) must be given, etc.
  - Did you use the correct function? CEIL, FLOOR, ROUND, and TRUNC all round numbers, but only ROUND and TRUNC can round to multiples other than 1, and all four round slightly differently (CEIL rounds up, FLOOR rounds down, ROUND follows regular rounding rules, and TRUNC drops the digits after the decimal point).

Problems with the Query Hanging, Looping, Returning Infinite Rows (Cartesian Joins)

- Queries that hang and return infinite rows are generally caused by bad joins (**Cartesian Joins**) created in the query, and generally indicates a needed join condition is missing. Cartesian products result in every row of one table being joined to every row in another table. This normally happens when no matching join columns (fields) are specified. For example, if table A with 100 rows is joined with table B with 1000 rows, a Cartesian join will return 100,000 rows. Cartesian products can negatively impact the system and possibly cause system crashes, ensure due diligence when creating queries that involve joins. Recheck each of your current join conditions and identify if additional join conditions are needed; sometimes asking someone to aid in the review with a fresh pair of eyes is helpful.

Problems with Unions

- Each query that is part of a union must have the same number of fields, same ordering, and must have consistent data types. If there is a difference, an error such as this one is displayed: "A UNION requires the same number of fields for each SELECT. Correct and retry."

Problems with Prompts

- A prompt that uses a prompt table may show no values when using the lookup feature. If the prompt table has a key with more than one field, there must be prompts for all fields before that field in the field order. For instance, if there is a prompt for Program (ACAD_PROG) using ACAD_PROG_TBL as a prompt table, since the key of ACAD_PROG_TBL is INSTITUTION + ACAD_PROG + EFFDT, there must also be a prompt for INSTITUTION. In addition, the prompts must be specified in the order of the fields of the key; in the example, the prompt on INSTITUTION must come before the prompt on ACAD_PROG.

Problems with Aggregates

- You cannot use fields that are being aggregated as regular criteria. If you apply the Count aggregate function to STRM, you cannot also add the criterion STRM = '1930'. When you aggregate, you are no

longer dealing with individual rows, but groups. (You can still add criteria on the fields that are not being aggregated.)

- Remember that once you add an aggregate to a displayed field, the aggregation is done on groups defined by the other fields. For example, if you want to count how many students there are per program, don't include EMPLID in the fields being displayed, or else the count will be per combination of program and EMPLID.
- If you put an aggregate function into an expression, you must check the Aggregate Function checkbox, or you will receive the error "not a single-group group function." This is indicating that you are trying to use an aggregate function on results that are not being aggregated.

Unable to find a query

- Make sure you are on the Query Manager Search Page (titled "Query Manager") rather than the Records tab (titled "Find an Existing Record"). Since both have search capabilities, one can mistake one of these search pages for the other page.
- If you tried searching for the full name of the query, you may have misspelled the name either when searching for it or saving it. Try entering less of the name. For instance, search for "%TRNG%" or even "TRNG" instead of "TRNG_QM##_Q15." You may get several extraneous results but have a better chance of getting the one that you need.

Look at the SQL tab!

- If you have a basic understanding of SQL, review the content of the SQL tab (and give learning basic SQL a try as well if it is unfamiliar to you). The query is not checked for errors before it reaches the database. The error messages reported by Query Manager are native to the database and refer to the generated SQL. To debug the query, it is sometimes easiest to look at the generated SQL. This will show which records are included, how they are joined (even if you didn't create the join conditions yourself); subqueries, effective date logic, and so on. It can also show a little more clearly which criteria are for which part of the query in the case of large queries.

Query doesn't Return Results to Excel (using IE Browser)

- Check that you have https://*.uh.edu included as a trusted site in your IE browser. If not navigate to IE → Tools → Options → Security → Custom Level, scroll down to the downloads section and make sure "Automatic Prompting for File Downloads" is enabled.

New/Upcoming Functionality

Recent upgrades to our Oracle PeopleTools framework will provide users with a more enhanced query reporting toolset. The functionality offered by these new features promises to be of great benefit to many users. The features are currently not fully functional yet for use, but will be incorporated in the Query toolset for use soon and training updated accordingly.

Below are the noteworthy enhancements, coming your way soon:

- **Drilling URLs:** When you build a query using Query Manager, you can define drilling URLs that are associated with this query as a column of output. When you execute this query through Query Manager or Query Viewer, the query results page shows results within that column as links, which you can click to be redirected to a different page in a new browser. Depending on how drilling URLs are defined, the new browser is either a PeopleSoft Pure Internet Architecture page, another query result page, or an external page. While defining a component drilling URL, you are able to select an action type to associate with the component. When you click the component drilling URL from the query results page, it opens the target component with the correct action mode. The available menu actions are Add, Update/Display, Update/Display All, and Correction.
- **PeopleSoft Pivot Grid:** A new reporting tool that enables you to visually display query data formatted as a Grid and/or Chart; the pivots also allow drill-down to details. If you view aggregate data—for example, SUM, AVG, COUNT, MAX, MIN—you can drill down to view the data that comprise the aggregate value displayed in the grid and chart by clicking a value in the grid or chart. You can display the grid from a chart-only view and then perform various actions—such as pivoting data, dragging and dropping, and slicing and filtering data—to change the grid layout. Pivot Grid Progressive Filtering is functionality provided in the Pivot Grid. The Query Manager component uses the Publish as Pivot Grid link to access the Review Pivot Grids page where users can add new Pivot Grid models or update existing ones that use the current query as the data source. Charting features include basic and enhanced functionality such as: Rating Boxes, gauges, ring charts, scatter and bubble diagrams, and smooth line and area charts.
- **Prompt Defaults and Optional Prompts:** Query prompts can now have default values, which are specified as constants or by the use of system variables such as %Date, %DateTime, %Time. Prompts can now also be set as optional.
- **Export More Than 64K Rows of Data to Excel:** PeopleSoft Query now supports Microsoft Excel 2007 and higher versions. As a result, you can send more than 64 thousand rows of query output to Microsoft Excel.
- **Image Support:** Queries can now include image fields in their result sets. Available options are to return embedded image data (base 64 encoded), image hyperlink, or display blank (legacy mode). If the image field property is set to Image Data, the image field value (binary image) will be displayed as an embedded image in the result set when you run the query to HTML output format. If you run a query to any other type of format (for example, XML or TXT), a Base64 encoded data string representing the image is displayed instead. When setting the property to Image Hyperlink, a URL will be returned in place of the field value. When accessed, this URL will return the image as an image file in a new browser window.
- **Find Query/Connected Query Definitional References:** PeopleTools 8.54 adds a Find References link, allowing a user to determine if a Query or Connected Query is used as a data source by another component, such as another Connected Query, BI Publisher report, Pivot Grid, Search Definition, or RSS Feed.

Assessment Exercises

ASSESSMENT EXERCISES

The below exercises facilitate assessment of attendees comprehension of the basic, intermediate and advanced query concepts presented during training. Each attendee will need to complete the exercises independently prior to receiving query write access in Production.

Assessment 1 – Basics:

Provide a list of juniors enrolled at UH-Main the last day of Fall 2008 that made the dean's list. The columns of output, in order, must include: Student ID, Academic Career, Academic Level, Academic Level Description, Academic Standing, Academic Standing Description, Cumulative GPA (the heading for the GPA should say "Fall 2008 GPA"), College, College Description, Primary Plan, and Primary Plan Description. The data should be sorted by academic standing code in ascending order, then by cumulative GPA in descending order. Save As "ASSESSMENT1_BASIC"; enter your PeopleSoft ID and last name in the query short description field.

Tip: you should receive 1,160 rows of data.

Assessment 2 – Intermediate/Advanced:

Provide a list that displays the EMPLID and CUM_GPA for all students enrolled at the end of the term during Fall 2009 at UH-Main if the student had a cumulative grade point average of at least 3.0. If the student belonged to the Honors (R003) activity type, provide that additional information in a column (UHIR_ACTIVITY_TYPE) as well for applicable students. Also add a calculated column that combines the student's current Academic Standing code with their Primary Plan (having a dash in between), i.e 'E-TELSBS'; the title of the calculated column should say "Standing and Plan". The institution should be a run-time variable that defaults to value 00730, having a lookup against master table 'INSTITUTION_TBL' and having custom heading text that says "Enter the Institution". Sort the data by activity type, then by student ID. Save As "ASSESSMENT2_INTERMEDIATE", enter your PeopleSoft ID and last name in the query short description field. Run the query for institution 00730.

Tip: you should receive 18,144 rows of data.

Contact Information

(UHS) Marie Coleman: macolem3@central.uh.edu ; (713) 743-9762

(UHCL) Cynthia Ustoy: ustoyca@uhcl.edu ; (281) 283-3273

(UHM) Susan Moreno: semoreno@central.uh.edu ; (713) 743-0640

(UHM) Carmen Allen: ceallenc@central.UH.EDU ; (713) 743-0642

(UHV) Mary Ann Rivera: riveram@uhv.edu ; (361) 570-4395