REPORTING UPDATE

9/7/10
Katherine Lindsay
College of Science & Engineering
OBJECTIVES:

- What tools do what?
- SQL overview
- Examples of queries
- Update on Business Intelligence Initiative
- Good, helpful resources
REPORTING TOOLS

TODAY...
• UMREPORTS
• REPORTING INSTANCE
• DATA WAREHOUSE (WEB QUERY TOOL)

TOMORROW... (END OF FY11? FY12?)
• OBIEE BUSINESS INTELLIGENCE TOOL (Oracle Business Intelligence Enterprise Edition)
UMREPORTS

• STANDARD REPORTS
• Tips & Tricks: http://blog.lib.umn.edu/itcomm/fiscaladmin/
• UMREPORTS will be going away...parallel plan in place
EFS REPORTING INSTANCE: QUERY MANAGER

- ABILITY TO RUN AD HOC QUERIES
- NEED QUERY MANAGER ACCESS TO CREATE QUERIES
- 2 DAY TRAINING CLASS
- 6 JOIN LIMIT (COMBINE TABLES)
- COPY FEATURE IS NICE BUT CUMBERSOME
- DATA DICTIONARY SHOWS ALL TABLES-IN PRODUCTION & REPORTING INSTANCE
Query Manager

This is a 2-day accelerated course for advanced users of the financial system. Due to the complexity of the query tool, the Controller’s Office and the Office of Budget and Finance identified competencies that are necessary to be successful in using this tool. Participants are expected to have a high level of knowledge, skills, and abilities for learning to use the Query tool and analyze the data results. To determine if this tool is right for you, review the [course description and required skill set (pdf)](#).

Prerequisites: [Basic Accounting, Chart of Accounts](#)
Also recommended: [Financial System Navigation](#)

Registration

Canceled Wednesday-Thursday, September 22-23, 8:30-4:00
Wednesday-Thursday, November 17-18, 8:30-4:00

[RRC manager approval is required for enrollment in this course.](#) Once Training Services has received an email with approval from the RRC manager, we will contact you about enrollment.
DATA WAREHOUSE

COLLECTION OF DATA ORGANIZED BY SUBJECT AREA

STUDENT ENROLLMENT (DWSA)

EFS FINANCIALS (DWEF)

RESEARCH ANIMAL RESOURCES (DWRA)

CUFS FINANCIALS (DWFS)

PAYROLL (DWPY)

STUDENT FINANCIALS (DWSF)

USTORES (DWUS)

GRADUATE SCHOOL (DWGR)
DWEF TABLE ACCESS VIA DW WEB QUERY TOOL

Asset Management
Account Receivable and Billing
Accounts Payable
Budgets
Chart of Accounts
General Ledger
Grants, Projects, and Contracts
Procurement
Travel and Expense
Treasury and Endowments
Today: Who uses what and why?

- Optimization: What’s the best that can happen?
- Predictive modeling: What will happen next?
- Forecasting/extrapolation: What if these trends continue?
- Statistical analysis: Why is this happening?
- Alerts: What actions are needed?
- Query/drill down: Where exactly is the problem?
- Ad hoc reports: How many, how often, where?
- Standard reports: What happened?


Slide taken from 7/1/10 Analyst BI presentation by OIT (Peter Radcliffe & Ben Cashen)
Reporting Instance vs. Data Warehouse

**MOST TRANSACTIONAL DATA IS IN DWEF**

SENSITIVE INFORMATION NOT SHOWN SUCH AS...
- CREDIT CARD NUMBERS
- SSNS/TAXPAYER IDS
- BANK INFORMATION
- HIPAA PROTECTED DATA

YOU MAY STILL NEED TO USE REPORTING INSTANCE DEPENDING ON BUSINESS QUESTION...
For example...PCARD data

Business scenario:
Need to ensure all PCARD transactions are reconciled/approved by 9/10/10. I’d like to give an ad hoc report to the preparer to show what is outstanding or has been reconciled on the default code by mistake so it can be corrected.

CENTRAL PUBLIC QUERY:
UM_DPO_PCARD_TRANS_BY_BARCODE

PROS: SEE WHAT HAPPENED, SHOWS MERCHANT, BRIEF DESCRIPTION, AMOUNT, DATE

PROBLEMS: PROMPT BY BAR CODE, HAVE TO DECIPHER EMPLID (CARDHOLDER), CS FIELD MISSING (BUDGET_REF FIELD MISSING), CAN’T SEE WHAT IS PENDING

TEST BARCODE: 1638091438107232010
For example...PCARD data (cont’d)

MODIFIED QUERY:
CSE_PCARD_TRANS (COPIED FROM UM_DPO_PCARD_TRANS_BY_BARCODE)
-REMOVED PROMPT
-INCLUDE CARDHOLDER NAME
-FLEXIBILITY TO INCLUDE BILLING DATE
-FLEXIBILITY TO SHOW VARIOUS ACCOUNT CODES (‘721101’)

DWEF Table
FS_PS_CC_ACCTG_LINE INTERFACE TO GL TABLES - ProCard Accounting Line

As of today...no pre-transaction data is available, just what has posted to the ledger.
DW TOOL

• BUILD A QUERY USING TOOL
• FREE-FORM QUERY USING SQL
Data Dictionaries

As defined in the IBM Dictionary of Computing... “centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format.”

RELATIONAL DATABASE MANAGEMENT SYSTEM (ORACLE)
DATA IS ORGANIZED BY ‘THEME’ INTO TABLES OF ROWS & COLUMNS

- REPORTING INSTANCE QUERY:
  UM_DATA_DICTIONARY
  **PROMPTS BY RECORD (TABLE NAME)**

- DW>INFORMATION>DATA DICTIONARIES>DWEF
  (NAME OF TABLE THAT HOLDS EFS FINANCIAL DATA)
Data Dictionaries (Cont’d)

WHAT FIELDS ARE AVAILABLE? WHAT TABLES SHOW WHAT?

2.2 INDIVIDUAL TABLE DESCRIPTIONS

2.2 Individual Table Descriptions

<table>
<thead>
<tr>
<th>FS_C_CDA_MAP_TEL</th>
<th>BBS CONVERSION TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS_PSTREEBRANCH</td>
<td>SETUP TABLES - Tree Branch</td>
</tr>
<tr>
<td>FS_PSTREEDEFN</td>
<td>SETUP TABLES - Tree Definition</td>
</tr>
<tr>
<td>FS_PSTREELEAF</td>
<td>SETUP TABLES - Tree Leaf</td>
</tr>
<tr>
<td>FS_PSTREETLEVEL</td>
<td>SETUP TABLES - Tree Level</td>
</tr>
<tr>
<td>FS_PSTREETNODE</td>
<td>SETUP TABLES - Tree Node</td>
</tr>
<tr>
<td>FS_PS_ASSET</td>
<td>ASSET MANAGEMENT - The primary Asset table populated when an asset is created</td>
</tr>
<tr>
<td>FS_PS_ASSET_ACQ_UET</td>
<td>ASSET MANAGEMENT</td>
</tr>
<tr>
<td>FS_PS_BI_ACT_ENTRY</td>
<td>BILLING - Accounting entry information for bills</td>
</tr>
<tr>
<td>FS_PS_BI_HDR</td>
<td>BILLING - Header record for Customer Bill</td>
</tr>
<tr>
<td>FS_PS_BI_HDR_NOTE</td>
<td>BILLING - Notes on Header record for Customer Bill</td>
</tr>
<tr>
<td>FS_PS_BI_LINE</td>
<td>BILLING - Line Information for customer bills</td>
</tr>
</tbody>
</table>

2.2 Individual Table Descriptions

FS_PS_BI_HDR

<table>
<thead>
<tr>
<th>#</th>
<th>COLUMN NAME</th>
<th>COLUMN_DESCR</th>
<th>TYPE/SIZE</th>
<th>TABLE SOURCE</th>
<th>COLUMN SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>business_unit</td>
<td>Business Unit</td>
<td>C(5)</td>
<td>ps_bi_hdr</td>
<td>business_unit</td>
</tr>
<tr>
<td>2</td>
<td>invoice</td>
<td>invoice</td>
<td>C(22)</td>
<td>ps_bi_hdr</td>
<td>invoice</td>
</tr>
<tr>
<td>3</td>
<td>bill_to_cust_id</td>
<td>bill_to_cust_id</td>
<td>C(1,5)</td>
<td>ps_bi_hdr</td>
<td>bill_to_cust_id</td>
</tr>
<tr>
<td>4</td>
<td>bill_status</td>
<td>bill_status</td>
<td>C(3)</td>
<td>ps_bi_hdr</td>
<td>bill_status</td>
</tr>
<tr>
<td>5</td>
<td>contract_num</td>
<td>contract_num</td>
<td>C(2,5)</td>
<td>ps_bi_hdr</td>
<td>contract_num</td>
</tr>
</tbody>
</table>
Business Rules:
Filters (Business Rules)

1) Fund Values
This report is limited to Current Non-Sponsored, Agency and Plant funds. Funds are determined by the PeopleSoft tree UM_FUND_CATEGORY, at the tree node CENTRALLY ALLOCATED, FO MONITORED, PLANT and AGENCY.

2) Financial Activity defined as:
Pre-Encumbrances/Encumbrances (Actualls) Table = PS_LEDGER_KK
Ledger = KK_DEX_ENC, KK_DEX_PRE
Business Unit = UMN01
Revenues/Expenditures (Actualls):
Table = PS_LEDGER
Ledger = ACTUALS
Business Unit = UMN01

3) Report results are restricted to accounts used by the Carryforward Process. Accounts used are those defined in the UM_CYFWD_ACCOUNTS tree at the following tree nodes:
REVENUES
EXPENDITURES
DEPRECIATION
PY_CYFWD_CASH
REVENUES_EXCL
PURCHASE_ACTUALS
INVENTORY
PREPAIDS
ACCR_LIABILITIES
TRANSFER_IN
TRANSFER_OUT
PY_CYFWD_ACCR
What is SQL? Structured Query Language, db computer language designed for managing data in relational db management systems.

Queries: Describe desired data and produce results

Most common way to do this is with ‘SELECT statement’. SELECT retrieves data from one or more tables, or expressions. SELECT * will pull all columns in the table.

*SQL is not case sensitive. SELECT is the same as select.

**SQL SYNTAX:**

SELECT clause: (column_name(s))
FROM (table_name)
WHERE (records that show specific criterion)

(SELECT *
 FROM DR_WHO_ACTOR
 WHERE ACTOR = ‘PETER DAVISON’)

= (equal to) > (greater than) < (less than) LIKE condition BETWEEN

GROUP BY ORDER BY HAVING IN AS

See handouts: DESKTOP REFERENCE: COMMONLY USED TABLES, SQL, SQL JOINS)
SQL QUERIES

• USE AS A BASE FOR FUTURE IDEAS / DEVELOPMENT
• LOTS OF WAYS TO GET SAME RESULT
• VALIDATE, VALIDATE, VALIDATE
  (UMREPORTS, SFR TRENDS, U-WIDE OFFICIAL REPORTS, OTHER ANALYSTS...)
*SQL is not case sensitive. SELECT is the same as select.

SQL SYNTAX:
SELECT clause: (column_name(s))
FROM (table_name)
WHERE (records that show specific criterion)

Example: Chart of Accounts Query

```sql
SELECT DESCR50 AS "CHART STRING NAME", FUND_CODE, DEPTID, PROGRAM_CODE, CHARTFIELD1, CHARTFIELD2, CHARTFIELD3 AS "EMPLID", PROJECT_ID, LASTUPDOPRID
FROM FS_PS_FGL_CF_MASK
WHERE DEPTID = '11055' and DESCR50 = 'INVALID CHARTSTRING'
```
EXAMPLE QUESTION: Which Chart strings are ‘invalid’?

<table>
<thead>
<tr>
<th>CHART STRING NAME</th>
<th>FUND_CODE</th>
<th>DEPTID</th>
<th>PROGRAM_CODE</th>
<th>CHARTFIELD1</th>
<th>CHARTFIELD2</th>
<th>EMPLID</th>
<th>PROJECT_ID</th>
<th>LASTUPDOPRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVALID CHARTSTRING</td>
<td>1000</td>
<td>11055</td>
<td>20330</td>
<td></td>
<td></td>
<td>1632963</td>
<td></td>
<td>PAPPONE</td>
</tr>
<tr>
<td>INVALID CHARTSTRING</td>
<td>1330</td>
<td>11055</td>
<td>20089</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PAPPONE</td>
</tr>
<tr>
<td>INVALID CHARTSTRING</td>
<td>1000</td>
<td>11055</td>
<td>20275</td>
<td></td>
<td></td>
<td>2103149</td>
<td></td>
<td>PAPPONE</td>
</tr>
<tr>
<td>INVALID CHARTSTRING</td>
<td>1000</td>
<td>11055</td>
<td>20259</td>
<td></td>
<td></td>
<td>3992309</td>
<td></td>
<td>PAPPONE</td>
</tr>
<tr>
<td>INVALID CHARTSTRING</td>
<td>1000</td>
<td>11055</td>
<td>20275</td>
<td></td>
<td></td>
<td>3310491</td>
<td></td>
<td>PAPPONE</td>
</tr>
<tr>
<td>INVALID CHARTSTRING</td>
<td>1000</td>
<td>11055</td>
<td>20275</td>
<td></td>
<td></td>
<td>3579609</td>
<td></td>
<td>PAPPONE</td>
</tr>
</tbody>
</table>

Note: You can change the column name to something useful. "Descr50" is less meaningful than "Chart String Name"

SQL: SELECT DESCR50 AS "CHART STRING NAME",  

BLANK DESCRIPTIONS (WHAT HASN’T BEEN NAMED...)

SELECT DESCR50 AS "CHART STRING NAME", FUND_CODE, DEPTID, PROGRAM_CODE, CHARTFIELD1, CHARTFIELD2, CHARTFIELD3 AS "EMPLID", PROJECT_ID, LASTUPDOPRID FROM FS_PS_FGL_CF_MASK WHERE DEPTID = '11055'
ORDER BY DESCR50

If you have 1 deptid, clause should be entered like this:
A.DEPTID = '11107'

If you have more than 1 deptid and they are sequential (i.e. 11101, 11102, 11103, 11104, 11105, 11106), clause can be entered like this:
A.DEPTID BETWEEN '11101' AND '11106'

If you have more than 1 deptid and they are not sequential (i.e. 11107, 11073), clause can be entered like this:
A.DEPTID IN('11107', '11073')
Tracking Payments from EFS Bills

SELECT B.CUST_ID, B.NAME1, A.INVOICE, TO_CHAR(A.INVOICE_DT,'YYYY-MM-DD'), A.INVOICE_AMOUNT,
TO_CHAR(C.ACCOUNTING_DT,'YYYY-MM-DD'), C.ENTRY_AMT, D.FUND_CODE, D.DEPTID, D.PROGRAM_CODE, D.PROJECT_ID,
D.CHARTFIELD1, D.CHARTFIELD2, D.CHARTFIELD3, D.ACCOUNT, C.PAYMENT_ID, A.BILL_SOURCE_ID, F.TEXT254,B.SETID
FROM FS_PS_BI_HDR A, FS_PS_CUSTOMER B, FS_PS_SP_SETID_CLSVW B1, FS_PS_ITEM_ACTIVITY C, FS_PS_BI_ACCT_ENTRY D,
FS_PS_BI_LINE_NOTE F
WHERE B.SETID = B1.SETID
  AND B1.OPRCLASS = 'UMALLPAGES'
  AND ( A.BUSINESS_UNIT = C.BUSINESS_UNIT
  AND A.INVOICE = D.INVOICE
  AND A.ITEM = C.ITEM
  AND C.ENTRY_TYPE = 'PY'
  AND D.ACCT_ENTRY_TYPE = 'RR'
  AND A.BILL_SOURCE_ID = '034'
  AND D.DEPTID = '11057'
  AND D.FUND_CODE = '1026'
  AND D.PROGRAM_CODE = '20102'
  AND D.ACCOUNT = '520204'
  AND A.BUSINESS_UNIT = F.BUSINESS_UNIT
  AND A.INVOICE = F.INVOICE )

Don’t forget to save your queries!

Save Your Query

Group Query By

Enter Query Description

Save Query

Group Katherine

Katherine 9/2/2010 4:43:26 PM PLTW_bill_payments_demo
EXPENSE REPORTS FOR MAIN + CHILD PROJECTS

```sql
SELECT C.ACCOUNTING_DT, A.SHEET_ID, A.MERCHANT, A.DESCRIPTION60, 
A.DESCRIPTION254, C.SHEET_NAME, C.EMPLID, D.NAME1, A.LINE_NUMBER, 
A.EXPENSE_TYPE, A.TRANSACTION_DATE, A.MONETARY_AMOUNT, 
B.DISTRIBUTION_LINE_NUMBER, B.BUSINESS_UNIT_ID, B.FUND_CODE, B.DEPT_ID, 
B.PROGRAM_CODE, B.PROJECT_ID, B.CHARTFIELD1, B.CHARTFIELD2, 
B.CHARTFIELD3, B.BUDGET_REFERENCE, B.ACCOUNT, B.BUDGET_DATE, 
C.OPR_ID_ENTERED_BY, C.BUSINESS_PURPOSE, C.TOTAL_REIMBURSEMENT_AMOUNT, 
D.SCHEDULED_PAYMENT_DATE, D.PAYMENT_METHOD 
FROM FS_PS_EX_SHEET_LINE A, FS_PS_EX_SHEET_DIST B, 
FS_PS_EX_SHEET_HDR C, FS_PS_EX_SHEET_PYMNT D 
WHERE A.SHEET_ID = B.SHEET_ID 
AND B.SHEET_ID = C.SHEET_ID 
AND C.EMPLID = D.EMPLID 
AND B.PROJECT_ID IN('00007584', '00007585', '02819009', '00014604', '00014605', '00014606') 
ORDER BY B.PROJECT_ID;
```
FROM FS_PS_VCHR_ACT_DRL_VW A, FS_PS_VOUCHER B, FS_PS_VENDOR C 
WHERE A.PROJECT_ID IN ('02819009','00007584','00007585','00014604','00014605','00014606') 
  AND B.BUSINESS_UNIT = A.BUSINESS_UNIT 
  AND B.VOUCHER_ID = A.VOUCHER_ID 
  AND C.VENDOR_ID = A.VENDOR_ID 
  AND B.ACCOUNTING_DT > TO_DATE('2010-06-30','YYYY-MM-DD')
DAVID OLSEN PAYROLL QUERY (MIRRORS PAYROLL TRANSACTION REPORT)

SELECT pay_end_dt as "action date", pay_period as "pay period", fisc_yr as "fiscal year", accounting_period as "accounting period", pay_end_dt as "pay end date", journal_id as "journal ID", fund, deptid_fin as deptid, account, nvl(trim(program_code),'XXXXX') as program, nvl(trim(project_id),'XXXXXXXX') as project, nvl(trim(chartfield1),'XXXXXXXX') as cf1, nvl(trim(chartfield2),'XXXXXXXXXX') as cf2, nvl(trim(chartfield3),'XXXXXXX') as finemplID, nvl(trim(budget_ref),'XX') as cs, jobcode as "job code", erncd as "earnings code", hours, (CASE WHEN substr(account,1,2)='70' then 'S' WHEN substr(account,1,2)='71' then 'F' ELSE null END) as "S/F", amount, name, emplid as ID, comprate as "comp rate" FROM PS_DWPY_PAY_FRNG_HIST_EFS_VW WHERE PROJECT_ID IN('00007584', '00007585', '02819009', '00014604', '00014605', '00014606') and fisc_yr='2011'
UNION ALL
SELECT entry_dt as "action date", aff_pay_period as "pay period", aff_fisc_yr as "fiscal year", accounting_pd as "accounting period", pay_end_dt as "pay end date", journal_id as "journal ID", fund, deptid_fin as deptid, account, nvl(trim(program_code),'XXXXX') as program, nvl(trim(project_id),'XXXXXXXX') as project, nvl(trim(chartfield1),'XXXXXXXX') as cf1, nvl(trim(chartfield2),'XXXXXXXXXX') as cf2, nvl(trim(chartfield3),'XXXXXXX') as finemplID, nvl(trim(budget_ref),'XX') as cs, jobcode as "job code", erncd as "earnings code", hours, (CASE WHEN substr(account,1,2)='70' then 'S' WHEN substr(account,1,2)='71' then 'F' ELSE null END) as "S/F", amount, name, emplid as ID, comprate as "comp rate" FROM PS_DWPY_HSA_HIST_EFS_VW WHERE PROJECT_ID IN('00007584', '00007585', '02819009', '00014604', '00014605', '00014606') and aff_fisc_yr='2011'
COST SHARE QUERY DRAFT
(2 QUERIES)

COST SHARE QUERY: PULLS BALANCES
SELECT A.FUND_CODE, A.DEPTID, A.PROGRAM_CODE, A.PROJECT_ID, A.CHARTFIELD1, A.CHARTFIELD2, A.CHARTFIELD3, A.BUDGET_REF, SUM(A.POSTED_TOTAL_AMT) AS "COST SHARE BALANCE"
FROM FS_PS_LEDGER A
WHERE A.FUND_CODE NOT BETWEEN '3000' AND '3999'
AND A.BUDGET_REF = 'CS'
AND A.LEDGER = 'ACTUALS'
AND A.DEPTID = '11055'
AND A.ACCOUNT NOT IN ('900101','900102')
ORDER BY A.PROJECT_ID;

COST SHARE QUERY: PULLS GRANT INFO
SELECT A.CONTRACT_NUM, TO_CHAR(A.BEGIN_DT,'YYYY-MM-DD'), TO_CHAR(A.END_DT,'YYYY-MM-DD'), A.TITLE56, A.NAME, A.DEPTID
FROM PS_GM_AWARD A
WHERE A.END_DT < TO_DATE('2010-04-01','YYYY-MM-DD')
AND A.DEPTID = '11093'
Tomorrow: OBIEE ‘BUSINESS INTELLIGENCE’

Oracle Business Intelligence Enterprise Edition

- A methodology for understanding the University’s operations based on data and analysis, not intuition.
- **Set of tools used in collection, integration and analysis of data.**

---

**Oracle BI Server**

- **Common Enterprise Information Model**
- **BI Interactive Dashboards**
- **BI Answers Ad hoc Analysis**
- **BI Delivers**
- **Microsoft Office Plug In**
- **Reporting & Publishing BI Publisher**

---

**OLTP & ODS Systems**
- Data Warehouse
- Data Mart

**Oracle PeopleSoft, Siebel, Custom Apps**

**Files Excel XML**

**Business Process**

**Other Local/Enterprise Systems**

---

*Slide taken from 7/1/10 Analyst BI presentation by OIT (Peter Radcliffe & Ben Cashen)*
OBIEE: Data

1. Pre-built warehouse with more than 16 star-schemas designed for analysis and reporting on Financial Analytics.

2. Pre-built ETL to extract data from hundreds of operational tables and load it into the DW, sourced from Peoplesoft.

3. Pre-mapped metadata, including embedded best practice calculations and metrics for Financial, Executives & other Business Users.
   - Presentation Layer
   - Logical Business Model
   - Physical Sources


Slide taken from 7/1/10 Analyst BI presentation by OIT (Peter Radcliffe & Ben Cashen)
Modeling & Analytics in the BI Continuum

- Strategic
  - •Answers
  - •Dashboards
  - •Cubes (OLAP)

- Dynamic
  - •Answers
  - •Dashboards
  - •Cubes (OLAP)

- Past Oriented
  - •Web Query Tool
  - •Gateway Access
  - •PS Query
  - •Answers

- Future Oriented
  - •EPM
  - •PS Data
  - •DW Data
  - •College Data

- Standardized Reporting
  - •UMReports
  - •PS Reporting
  - •BI Publisher
  - •Answers

- Ad-Hoc Query & Reporting
  - •PS Data
  - •DW Data
  - •College Data
  - •EPM

- Operational
  - •PS Dt

- Static

Slide taken from 7/1/10 Analyst BI presentation by OIT (Peter Radcliffe & Ben Cashen)
FREE SQL RESOURCES

Data Warehouse: https://dw.umn.edu/maininfo.asp

Click here: http://www.lib.umn.edu/libdata/page.phtml?page_id=1299

Scroll down to "Safari Books Online" which has been paid for and licensed by UMN. Type "SQL" and see the myriad of free e-resources we have access to in the collection.

Structured Query Language (SQL) Syntax:

SQL Joins/Unions:

HOW DO YOU GET ACCESS?

ARF FORM: http://www1.umn.edu/datasec/security/ARF_Instructions.htm

FINANCIAL DW TABLES (DWEF, DWFS):
http://www1.umn.edu/datasec/security/documents/2-FinancialDataWarehouse.doc

QUERY MANAGER (PLUS 2 DAY TRAINING):
QUERY EXCHANGE PILOT

• OCTOBER 15TH 1-2pm (ME 302)
  DETAILS TBD