



United States Department of Agriculture
US Forest Service
Natural Resource Manager (NRM)



FSVeg Common Stand Exam – Querying FSVeg

June 2015



TABLE OF CONTENTS

Overview	3
Part 1: PL/SQL Developer & Command Line Authenticator	3
Filtering Objects	8
Built-In Aids	12
Templates.....	14
Using a Template	17
Exercise 1	18
To Run a Query	18
Part 2: Basic SQL Query Format.....	18
Exercise 2	18
Exercise 3	19
Exercise 4	19
Use of Count and Sum with Group By	19
Exercise 5	19
Exercise 6	19
Part 3: FSveg Table Structure.....	19
Part 4: Developing Queries	23
Retrieving Stand, Plot, and Tree Data Tables and Columns.....	23
Exercise 7	24
Exercise 8	24
Exercise 9	24
Exercise 10	24
Retrieving Stand, Sample Designs, and Selection Criteria Tables and Columns.....	24
Exercise 11	24
Exercise 12	25
Exercise 13	25

Exercise 14	25
Subquery: Most Recent Measurement	25
Exercise 15	26
Exercise Answers	27
Exercise 1	27
Exercise 2	27
Exercise 3	27
Exercise 4	27
Exercise 5	27
Exercise 6	28
Exercise 7	28
Exercise 8	28
Exercise 9	28
Exercise 10	28
Exercise 11	29
Exercise 12	29
Exercise 13	29
Exercise 14	29
Exercise 15	30
Part 5: Examples From the Real World	30
Single-Table Queries	30
Query 01 – Number of stands and sum of acres by project name	31
Query 02 – Number of Stands and Sum of Acres by Project Name and EV_CODE	31
Multi-Table Queries	32
Query 03 – Join Tables	33
Views.....	35
Things to Keep in Mind	35

OVERVIEW

1. Query Development Environment: PL/SQL Developer
2. Basic Query Format
3. FSVeg Table Structure: Primary Components
 - a. Data: stand, plot, tree, etc.
 - b. Metadata: sample designs, selection criteria
4. Developing queries
5. Examples From the Real World

PART 1: PL/SQL DEVELOPER & COMMAND LINE AUTHENTICATOR

Simple and fast [SQL development environment](#).

Figure 1: Get PL/SQL Developer

Step 1. Go to this web page

Step 2. Go to this link (see the illustrations on the next page).
a. Follow the **PLSQL Current Version** link
b. Copy the product code, serial number, and password
c. Select **plsqlidevxxx.exe** to install it

DON'T mess with the add-ons.

PL/SQL Developer - Windows Internet Explorer provided by USDA Forest Service

http://fsweb.nris.fs.fed.us/For_Developers/plsqdev.shtml

File Edit View Favorites Tools Help

NRM Calls Google HP RAM info Live Meeting My Work Tracker GeoMAC (fire) FSVeg DEVN TEST PROD SF Artifacts

PL/SQL Developer TeamForge : List Documents

FSWeb - USDA Forest Service Intranet

NRM FSWeb | EMC FSWeb | WO FSWeb | USDA Intranet | NRM WWW | FS WWW | USDA WWW

Natural Resource Manager

Natural Resource Information System

NRM Help Desk | FACTS | Infra | NRIS | TIM

Login to NRM Login to Citrix Search NRM

You are here: Home > For Developers > PL/SQL Developer

PLSQL Developer

PLSQL Developer is a tool developed by Allround Automations. You can read a [brief overview](#), a [review](#), and a [comparison table](#) on the Allround Automations website regarding the PL/SQL Developer tool.

In acquiring enough licenses to cover NRIS users, the point was reached where it became cost effective to get an unlimited license for the agency. This software is licensed for all people working for the USDA Forest Service. This page should not be presented outside of the agency firewall. If you are not working for the USDA Forest Service, you should not be accessing this page.

Installation

1. To install and use the software, **Lounge**. Note the "Package Definition" for PLSQL Developer.
 - [Get PLSQL Developer](#)
2. Run the file to install it. If you have a product code, you will be prompted for a product code. Use the current license information as noted above.

Add-Ons

We strongly recommend that you download and install the Browser Extender ([be.zip](#)). This add-on enhances the functionality of the basic tool and even allows you to create your own new functionality. See the [Add-Ons page](#) for additional items.

Figure 2: Link on TeamForge

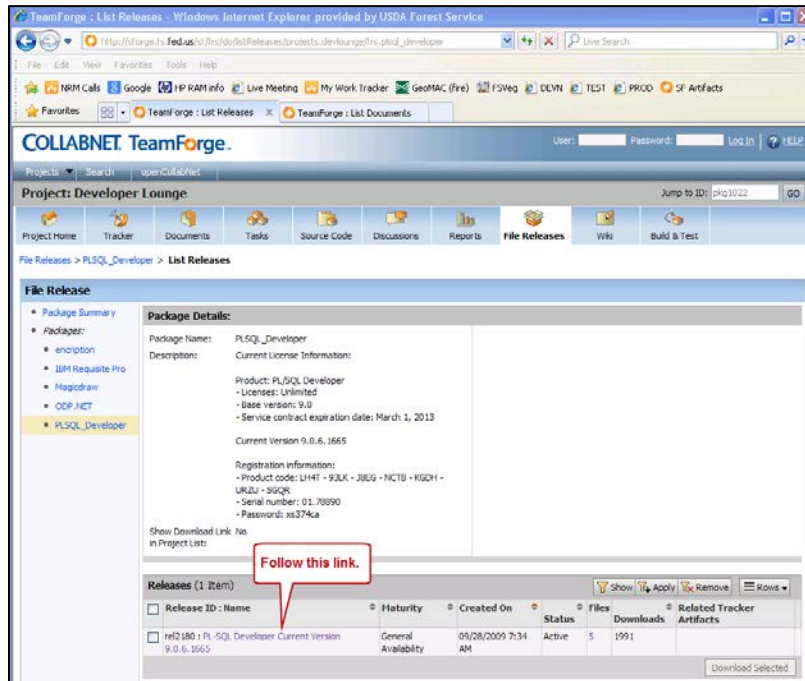
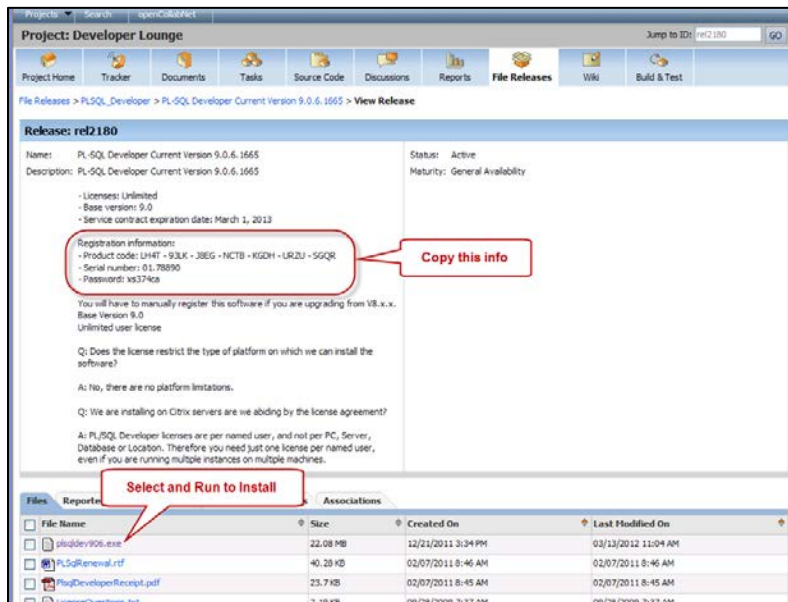


Figure 3: PL/SQL content to copy¹

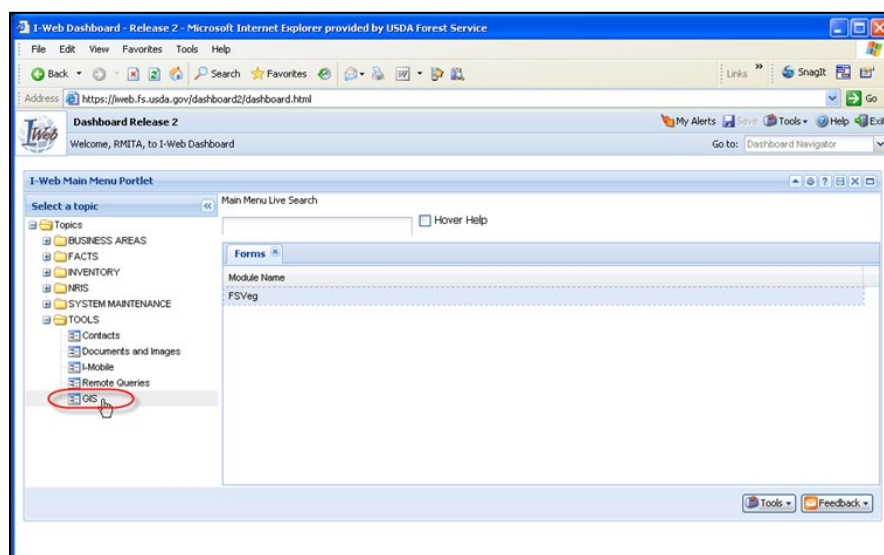


¹ The PL/SQL Developer version number will change over the years – don't worry if what you see isn't exactly as shown in these screen captures.

Once PL/SQL Developer is installed, you must obtain and install the Command Line Authenticator (CLA). There are two ways to do this. If you can, go to the [NRM Command-Line Authenticator](#) website and read over the requirements, then download and install the CLA via Run Elevated. If you can't access the site directly, follow these steps:

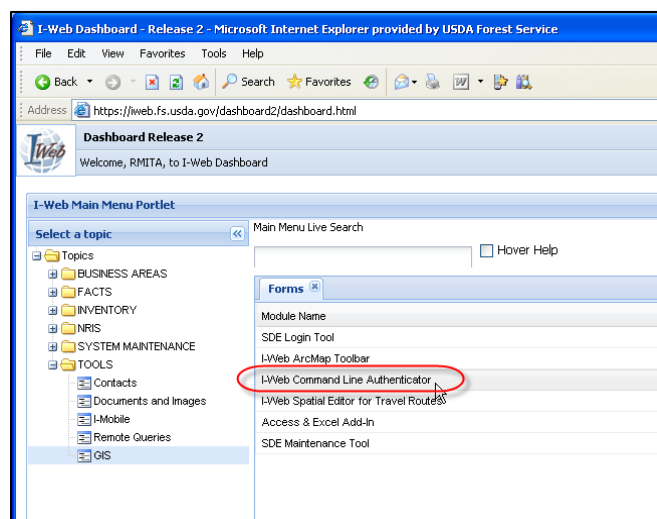
1. Go to the I-Web site
2. Select Default I-Web Dashboard
3. Go through the e-Authentication process; you will end up at the I-Web Dashboard, as seen in [Figure 4](#).
4. Select, from the menu tree on the left side of the screen: **Tools / GIS**

Figure 4: I-Web Dashboard

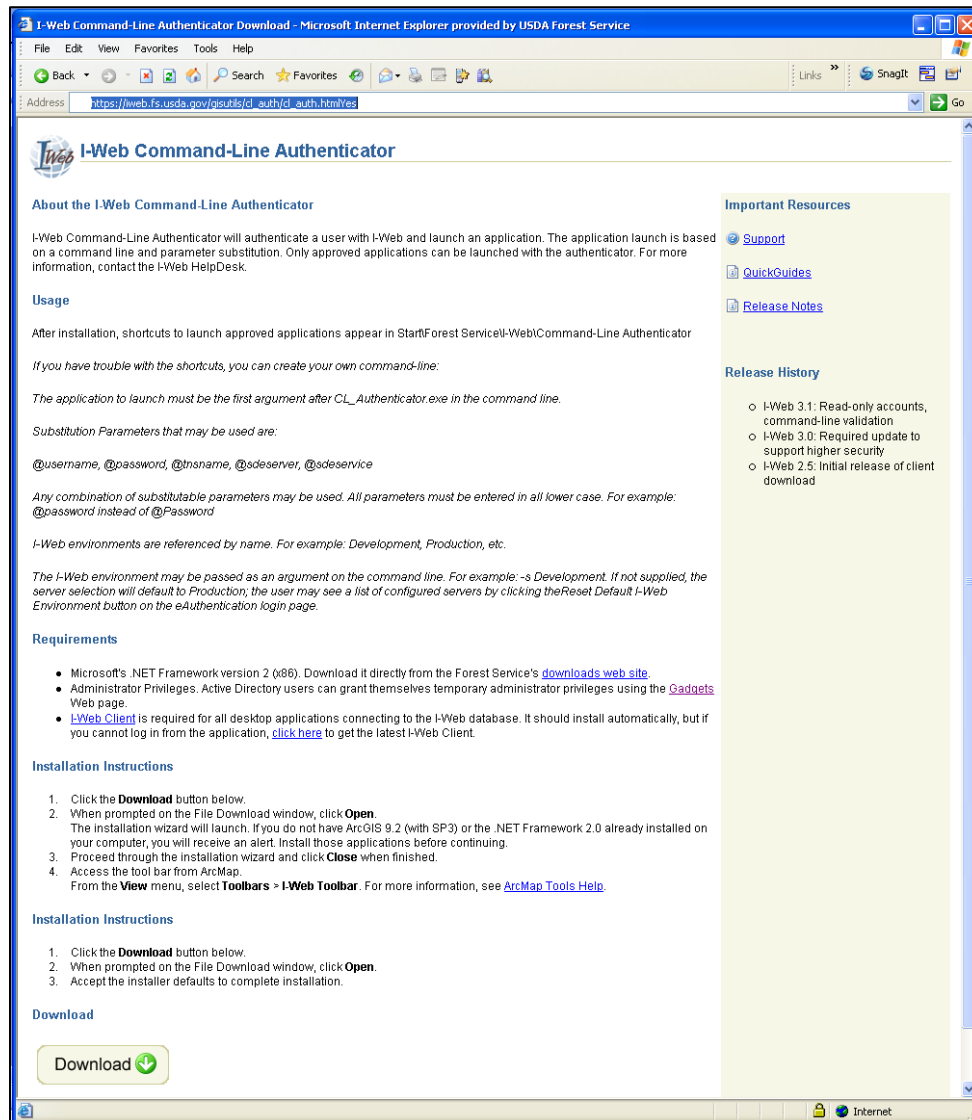


5. Select the I-Web Command Line Authenticator

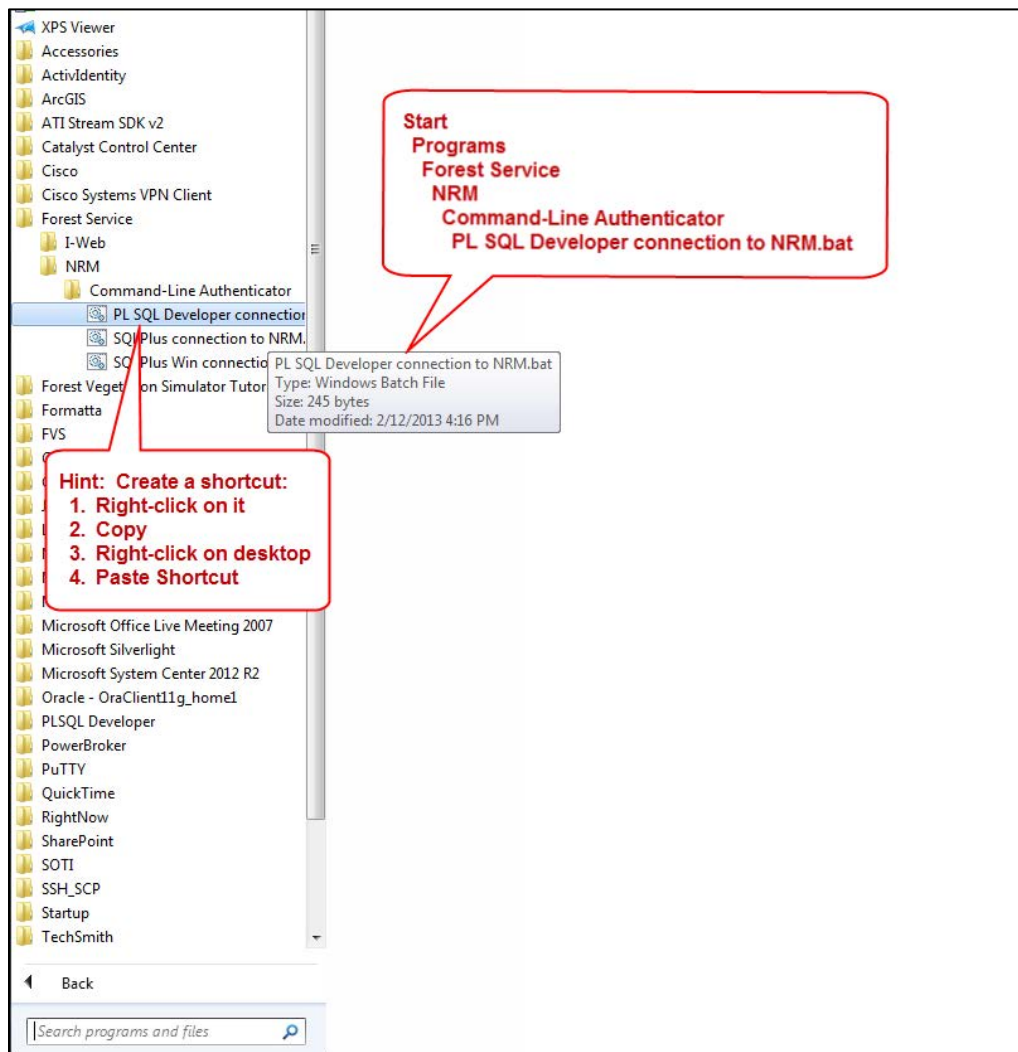
Figure 5: Access the I-Web Command Line Authenticator



6. Read over the requirements for the Command Line Authenticator, download it, and install it via Run Elevated.



Once installed, find it on your machine by doing the following:



Create a shortcut that you can place on your desktop or in your Start menu so that you don't have to do all this navigation every time you wish to run queries.

To start a PL/SQL Developer session, select the "PL SQL Developer connection to I-Web" icon (**DO NOT select the PL/SQL Developer icon directly**). It will take you through the e-Authentication and fire up a PL/SQL Developer session in which you can query the FS Veg data at NITC.

FILTERING OBJECTS

Figure 6: Open the Filters

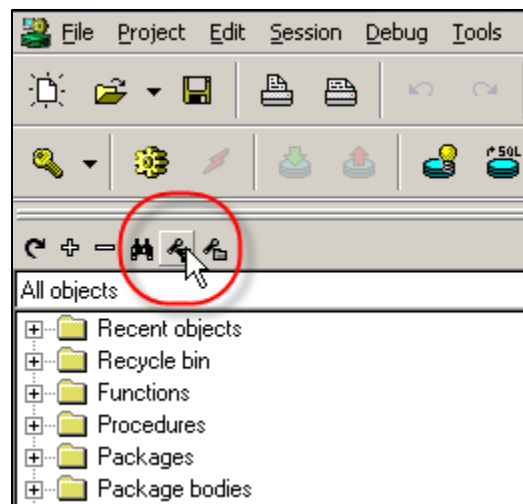


Figure 7: Create a New Filter

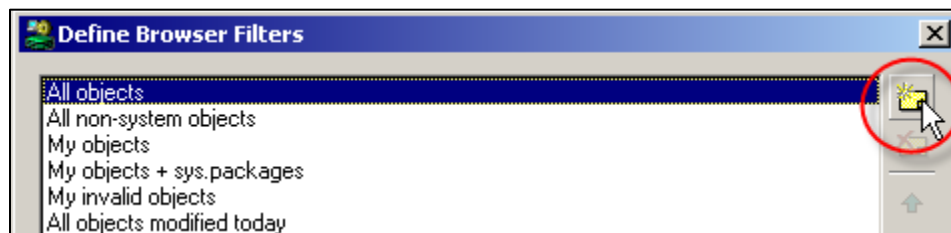
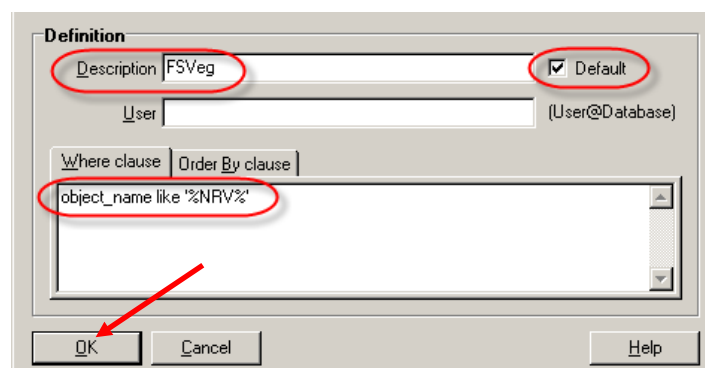


Table 1: FSVEg Roles, Descriptions, Permissions

Description	WHERE clause
FSVeg	object_name LIKE '%NRV%'
FSVeg Spatial	bject_name LIKE '%NRIS%APRV'

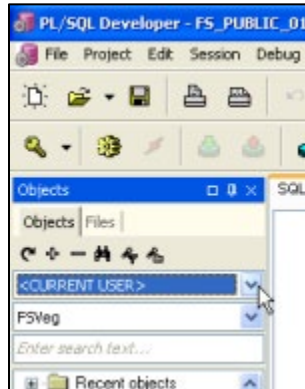
Figure 8: Set FSVeg or FSVeg Spatial as your Default Filter; click OK



Once the filter is built, enable the appropriate “Selected User” (which I think means “schema” for us) by doing the following:

1. Select the drop-down list button for the Selected User field

Figure 9: Selected User field



2. Very quickly type the first two characters of the desired Selected User; this will rapidly move your position in the list close to your desired value
3. Select the desired Selected User from the list

Figure 10: Accessing FS_NRIS_FSVEG (Selected User) Object



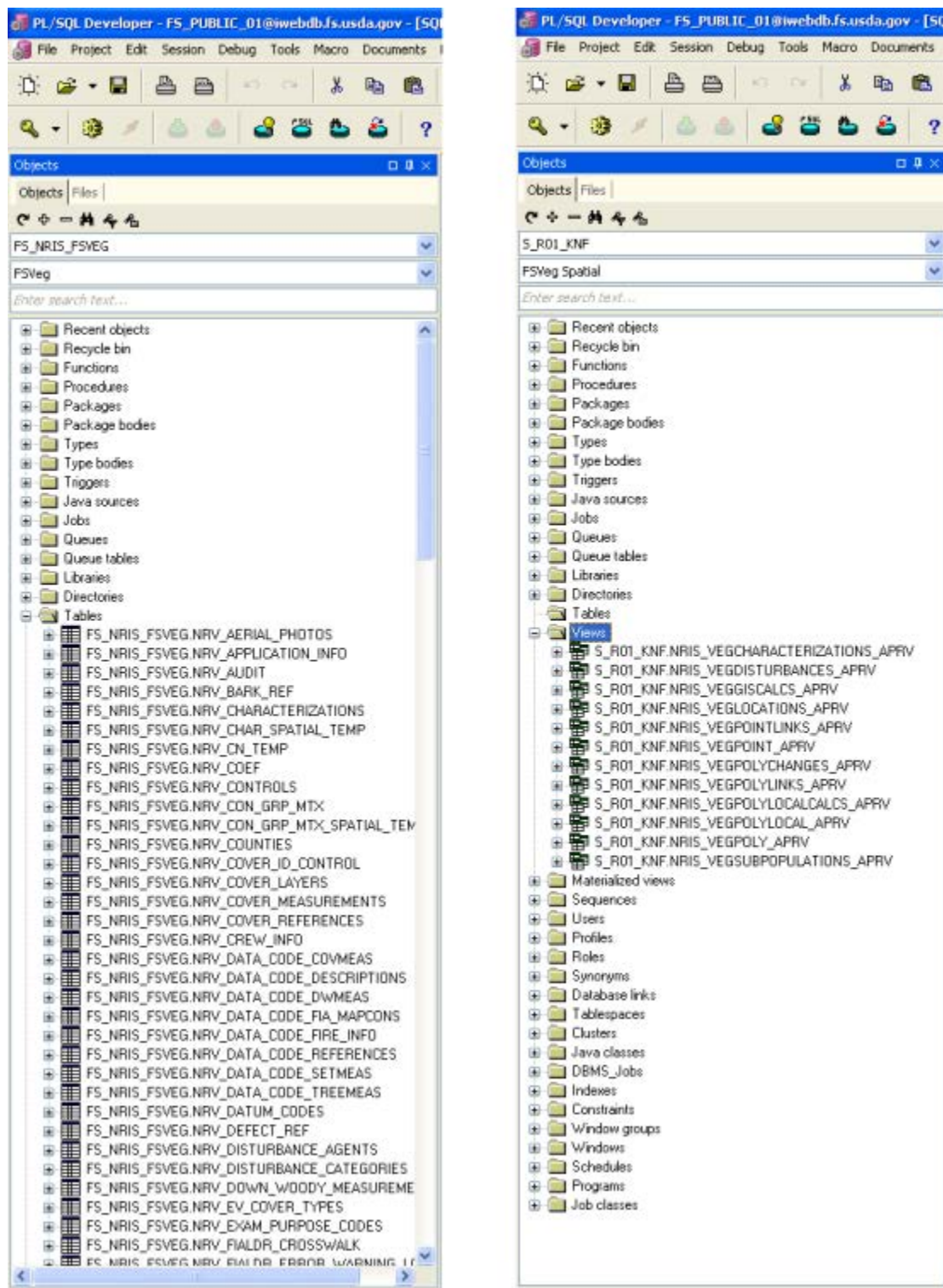
Table 2: Selected users by application

Application	Selected User
FSVeg	FS_NRIS_FSVEG

Application	Selected User
FSVeg Spatial	S_R<2-digit Region>_<2- to 4-digit Forest> (e.g., S_R01_KNF for the Kootenai NF)

When the filter has been created and the appropriate Selected User has been chosen, you will be able to view the items as desired by category (e.g., tables, views, etc.) within that schema.

Figure 11: FSVEg (left) & FSVEg Spatial (right) examples



BUILT-IN AIDS

Figure 12: Keep track of multiple queries

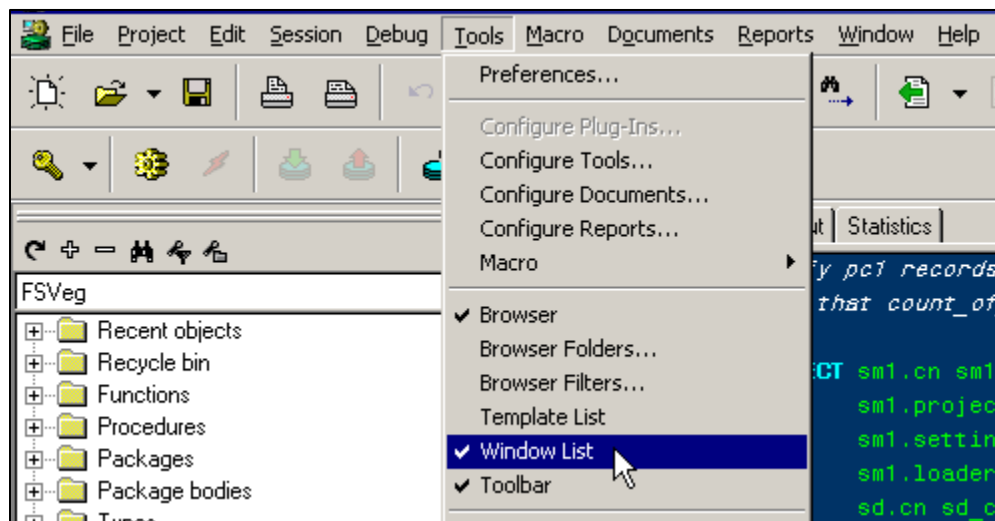


Figure 13: Columns for tables and views

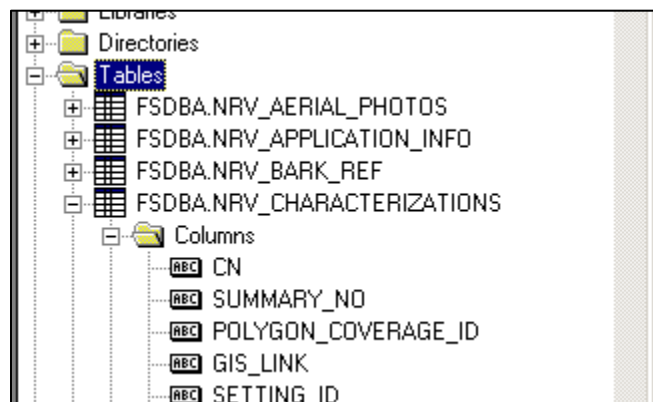


Figure 14: Table autofill

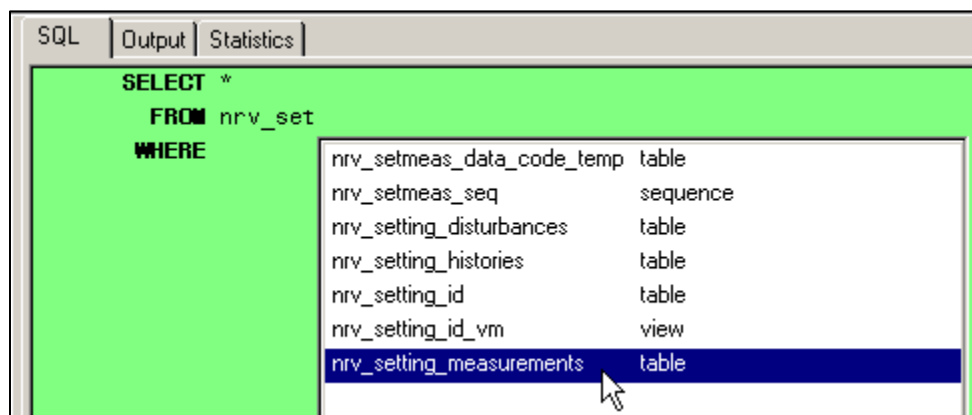


Figure 15: Column autofill (use table aliases)

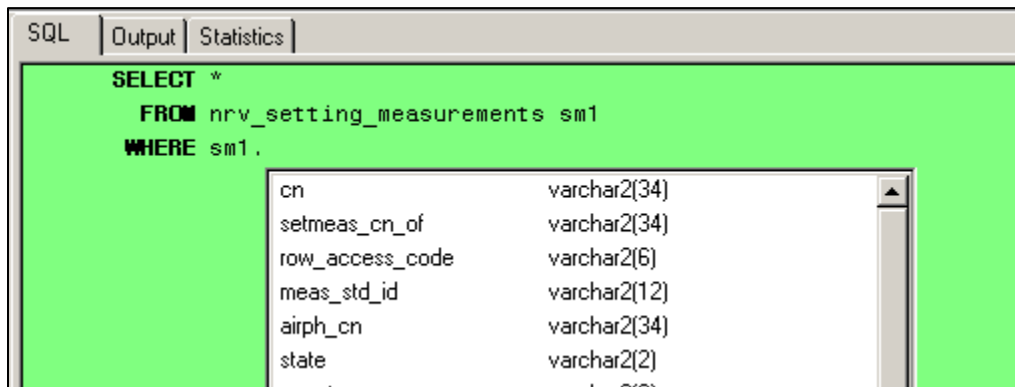


Figure 16: Obtain just a page of data, or all of it

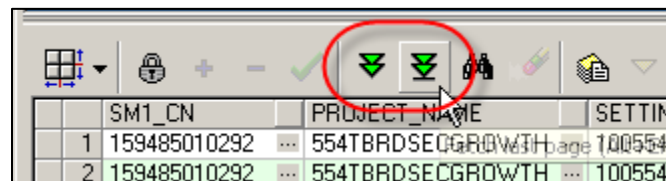
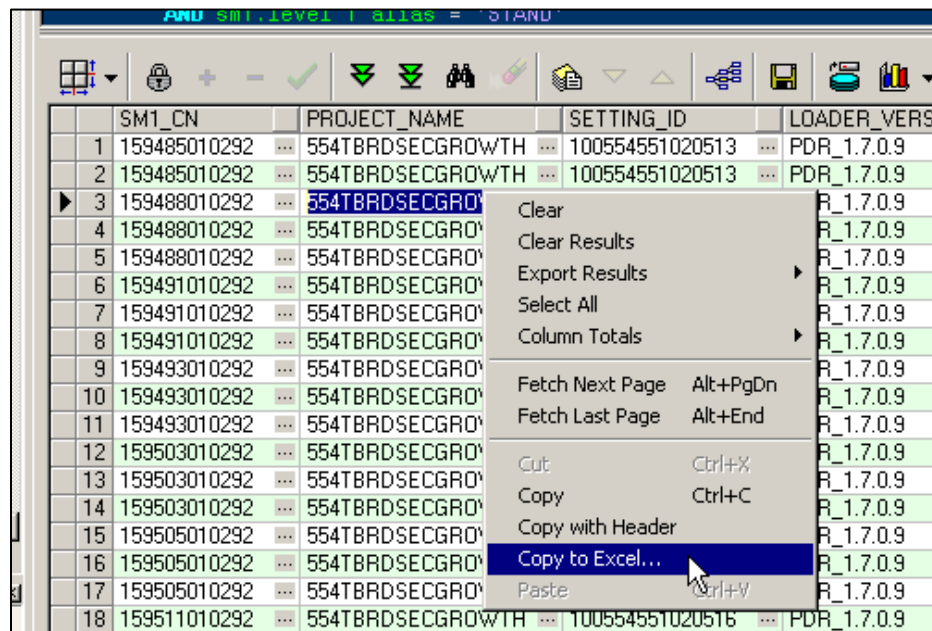


Figure 17: Copy results to Excel



TEMPLATES

Figure 18: Showing the Template List

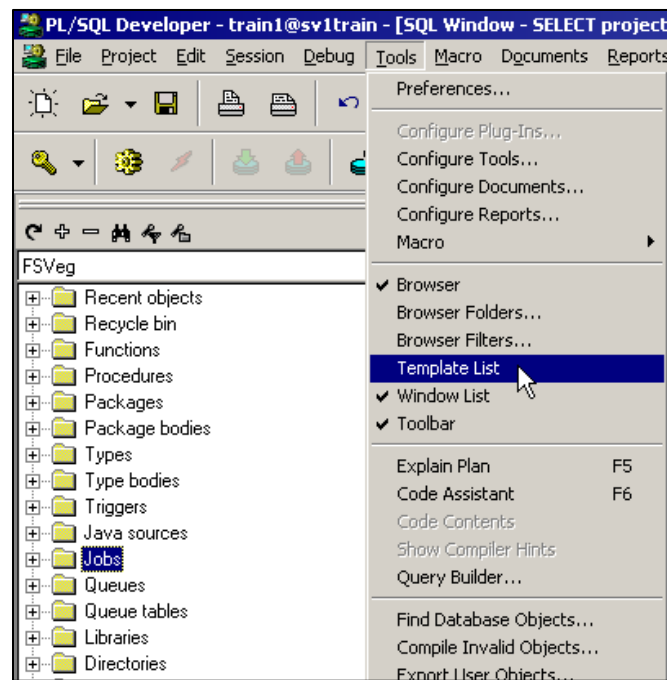


Figure 19: Template list

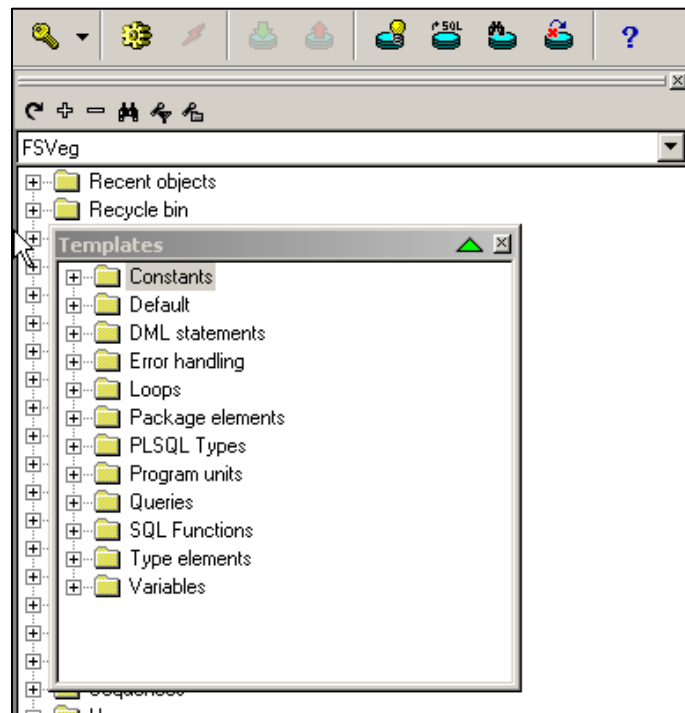


Figure 20: Move Template window

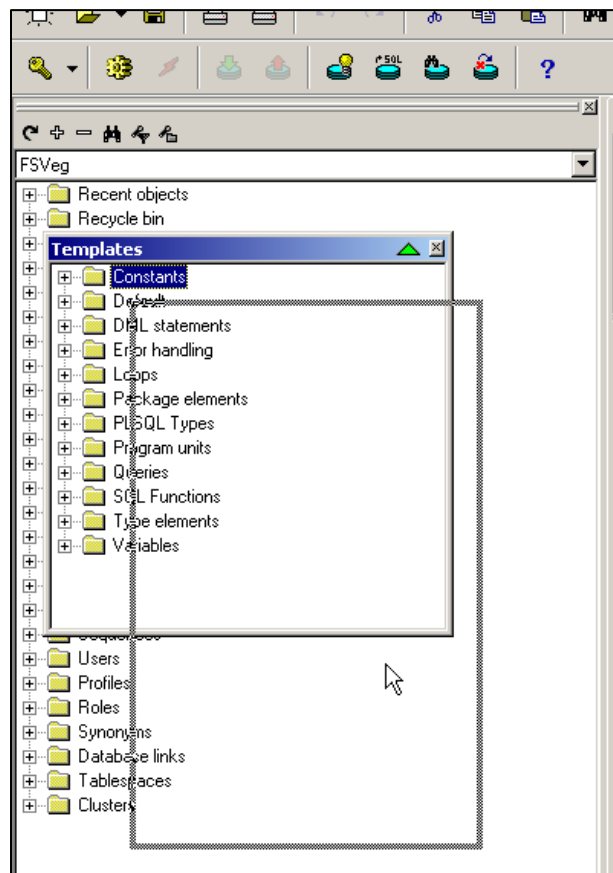


Figure 21: Template lists repositioned as a tab

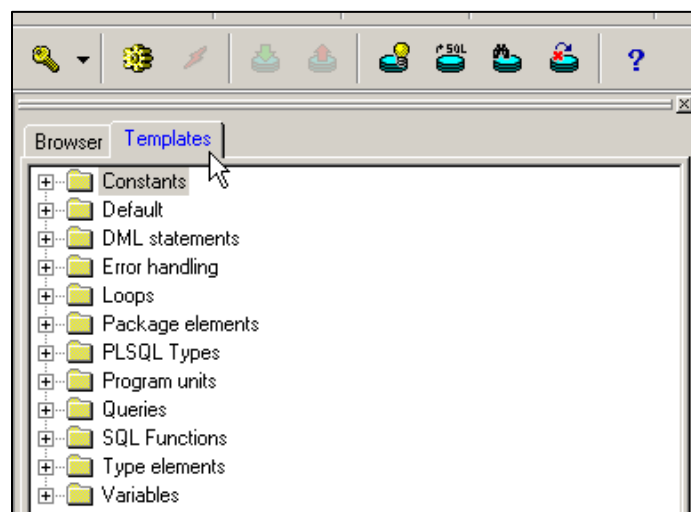
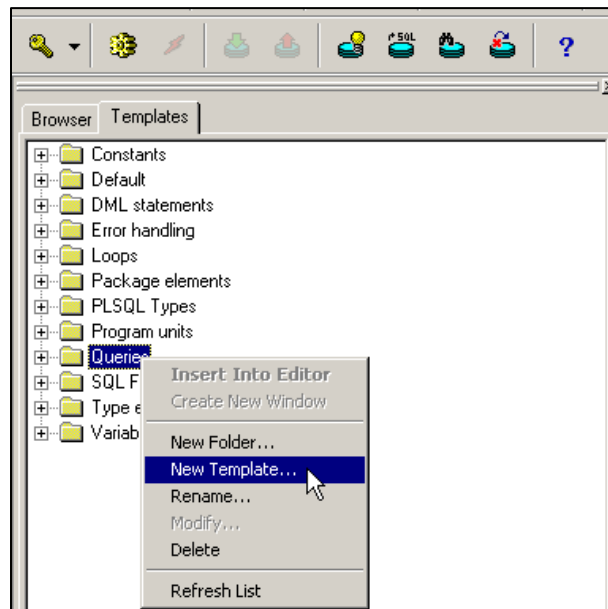


Figure 22: Create a new template



(Note: If the folder “Queries” doesn’t exist, create it by right-clicking in the template area and selecting New Folder.)

Figure 23: Enter new template name and click OK

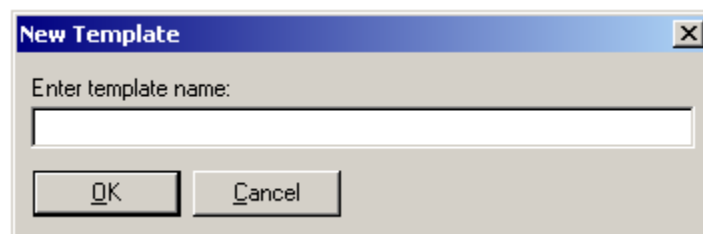
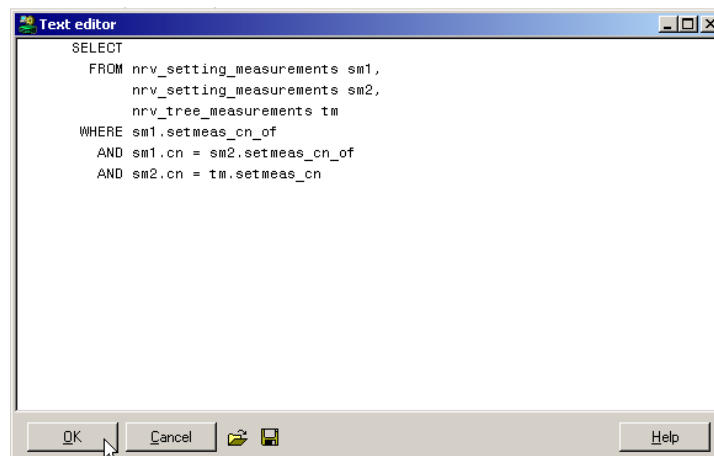


Figure 24: Text editor window



USING A TEMPLATE

Figure 25: Step 1 – Open an SQL window

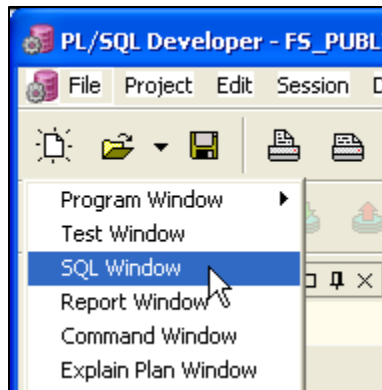
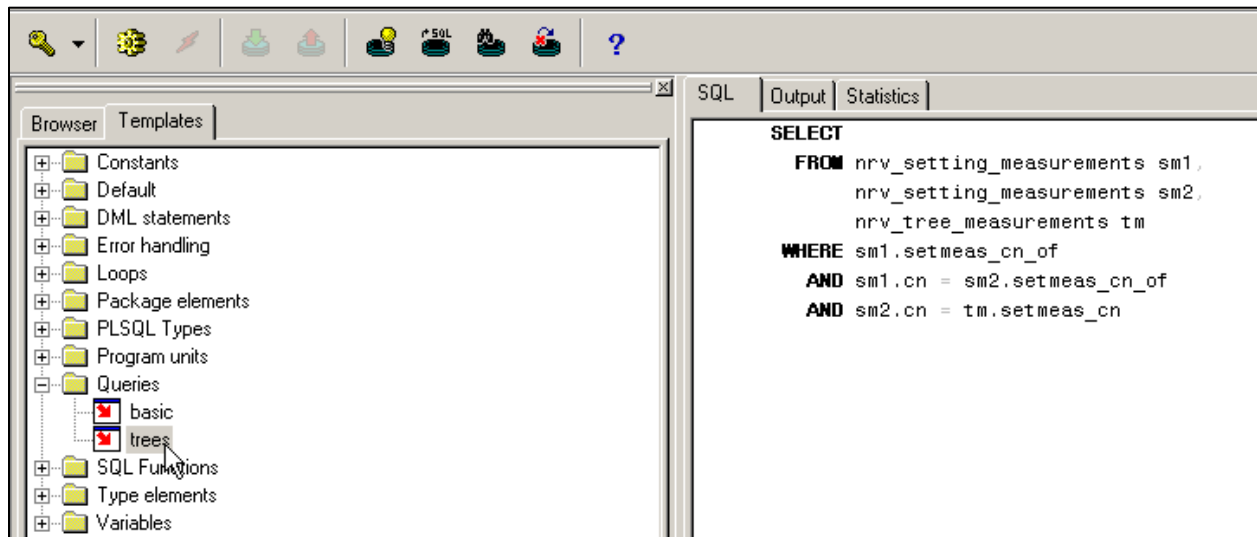


Figure 26: Step 2 – Navigate to and Select Template



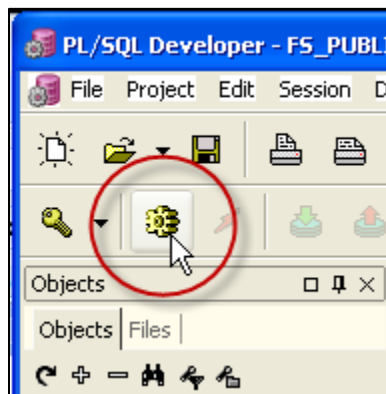
EXERCISE 1

Create an FSveg Filter and Make it Your Default

TO RUN A QUERY

1. Enter a query of your own creation into the SQL window or open a query that you have stored (it will open its own SQL window)
2. Select the Execute button (it looks like a little yellow cog in the upper left part of the PL/SQL Developer window)

Figure 27: Execute button



3. If the query is valid, it will run and the results will appear in the lower half of the SQL window; if there is a problem, the software will attempt to identify the issue as well as indicate where the problem exists by placing the cursor on the offending line

PART 2: BASIC SQL QUERY FORMAT

The basic parts of a query include:

```
SELECT <column 1>,  
        ...  
        <column n>,  
FROM   <table1>,  
        ...  
        <table m>  
WHERE  <however you want to subset the data>  
ORDER BY <column x>
```

EXERCISE 2

Open an SQL window in PLSQL Developer (New / SQL Window) and write a query that selects all project_name, setting_id, and measurement_date in nrv_setting_measurements where:

1. region_proc = '09'

2. forest_proc = '19'
3. level_1_alias = 'STAND' (to exclude FIA data)
4. setmeas_cn_of is NULL

EXERCISE 3

How many settings did you get?

EXERCISE 4

Modify your query so that the results are obtained by project_name and setting_id, in that order. You'll need to use the ORDER BY clause.

USE OF COUNT AND SUM WITH GROUP BY

```
SELECT <columns of interest>,  
        COUNT(<column x>),  
        SUM (<column y>)  
FROM <table1, table2, ..., tableN>,  
WHERE <however you want to subset the data>
```

EXERCISE 5

Copy your query to a new SQL window and modify it so that it now selects project_name, the number of setting_ids, and the sum of acres (setting_size) in those settings for each project_name. Use COUNT, SUM, and GROUP BY in your modified query. Order the results by the sum of acres.

EXERCISE 6

Copy your results to MS Excel. Re-order the results by project_name, and make a simple bar chart of the results.

PART 3: FSVeg TABLE STRUCTURE

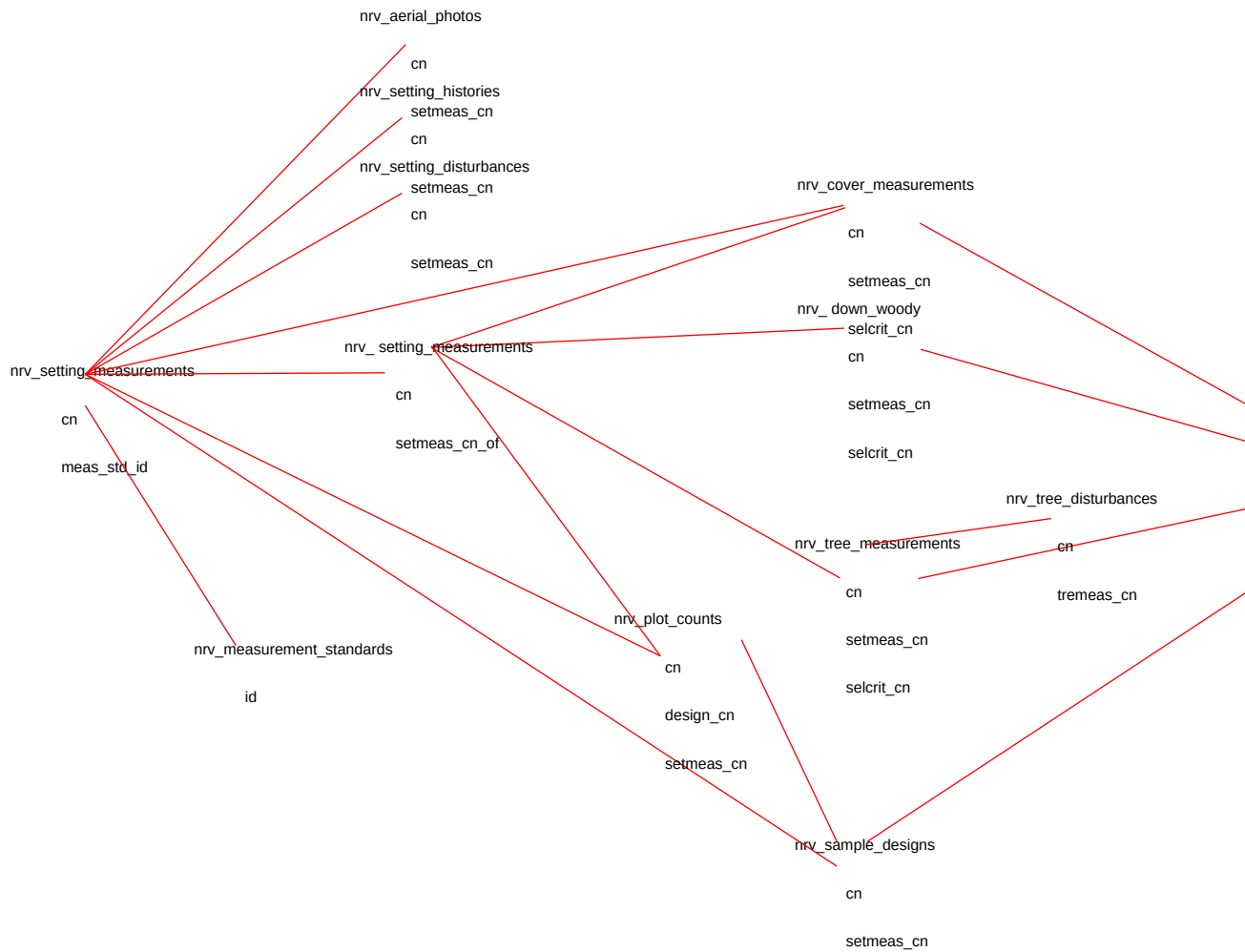
1. Informal "map" of selected FSVeg tables and relationships
 - a. Caveats
 - i. Not an entity relationship diagram (ERD)
 - ii. Not complete at all
 - iii. Not for surveys that have more than 2 levels
 - b. How to use it
 - i. Depending on your query strategy
 1. Which tables will you go after

2. Order of joins (smallest number of rows first – a production issue)
 - ii. Aid in writing joins
 1. Key and foreign key fields
 2. Recursion in nrv_setting_measurements
 3. Some tables can be joined to more than one level in nrv_setting_measurements; this helps keep that straight
 - iii. Start at an “anchor” table, then work your way around
2. CNs
 - a. VARCHAR2(34)
 - b. A pain at first, but they will save your sanity in the long run
3. Audit columns
 - a. created_by, created_date_in_instance
 - b. modified_by, modified_date_in_instance (last update only)
4. Collector version, loader version
 - a. Collector: paper forms or PDR version x?
 - b. Loader: PDR loader, data entry forms, legacy loader?
 - c. If data problem with either, can track it down
5. nrv_setting_measurements: level IDs and aliases

Table 3: FSVeg Roles, Descriptions, Permissions

Level	Stand Exam	FIA Grid
1	'STAND'	'CLUSTER'
2	'PLOT'	'PLOT'
3	n/a	'SUBPLOT' or 'TRANSECT'
4	n/a	'SUBSAMPLE'

6. nrv_setting_measurements: recursion; has both parent and child records
 - a. Parent setting ("stand"): setmeas_cn_of = NULL
 - b. Child setting ("plot"): setmeas_cn_of = parent setting cu
7. archive_flat in nrv_setting_measurements:
 - a. A survey is archived when:
 - i. Superseded by a newer survey
 - ii. Altered stand conditions render existing survey no longer relevant
 - iii. Stand redelineation
 - b. 'Y' denotes setting records that are no longer considered current
 - c. Can be used by queries and applications as a filter



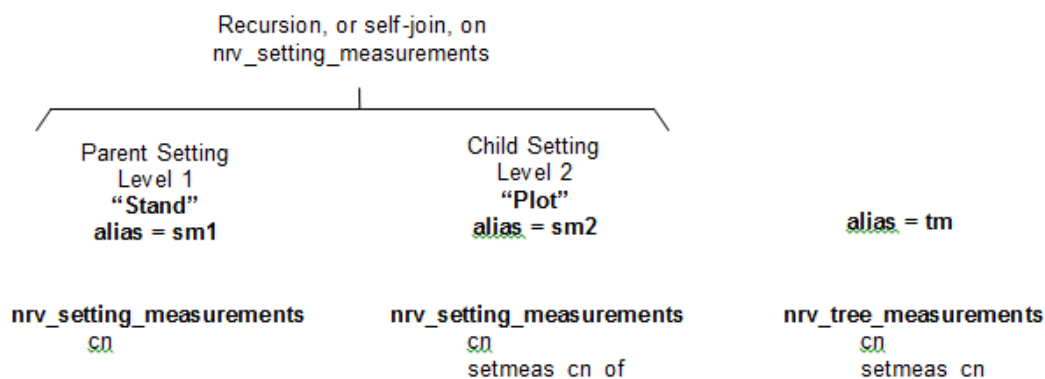
Map of selected FSveg tables and their relationships – note that only one recursion is shown for nrv_s

c:\my documents\rel_map_2.ppt 03-17-2004

PART 4: DEVELOPING QUERIES

RETRIEVING STAND, PLOT, AND TREE DATA TABLES AND COLUMNS

Figure 28: Process of retrieving data tables and columns



A query using this part of the map would have the basic components:

```
SELECT <columns 1>,  
      ...  
      <column n>  
FROM nr_v_setting_measurements sm1,2  
      nr_v_setting_measurements sm2,3  
      nr_v_tree_measurements tm4  
WHERE sm1.setmeas_cn_of IS NULL  
      AND _____5  
      AND _____6  
      ...  
      AND (sm1.region_proc = '09' OR sm1.region_admin = '09')7  
      AND (sm1.forest_proc = '19' OR sm1.forest_admin = '19')
```

² "Stand"

³ "Plot"

⁴ "Tree"

⁵ Stand-to-Plot join

⁶ Plot-to-Tree join

⁷ Make sure you narrow down your search with something like these two lines, since the entire nation's data is at NITC

EXERCISE 7

Connect the columns you would use to join the three tables.

EXERCISE 8

Fill in the blanks above to enable the table joins.

EXERCISE 9

Create a template for this query in PL/SQL Developer

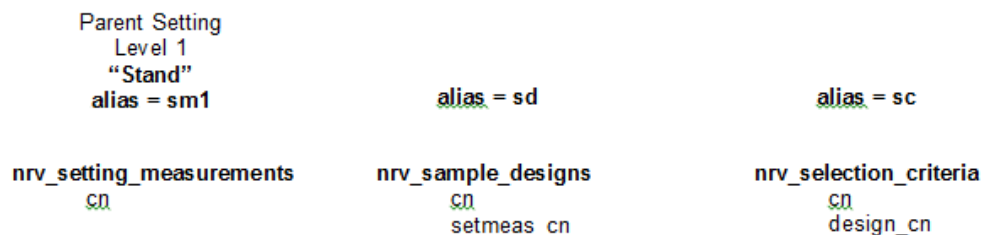
EXERCISE 10

Use the template to create a query that produces sm1.project_name, s1.setting_id, sm.measurement_date, sm2.level_2_id, tm.live_dead, tm.species_symbol, and tm_height, where:

1. project_name = 'BROOKSTON'
2. setting_id = '091902007040003'

RETRIEVING STAND, SAMPLE DESIGNS, AND SELECTION CRITERIA TABLES AND COLUMNS

Figure 29: Retrieving stand, sample designs and selection criteria



This query would have the basic components:

```
SELECT <columns 1>,  
      ...  
      <column n>  
FROM   nrv_setting_measurements sm1,  
        nrv_setting_measurements sm2,  
        nrv_tree_measurements tm  
WHERE  sm1.setmeas_cn_of IS NULL  
      AND _____  
      AND _____  
      ...  
      AND (sm1.region_proc = '09' OR sm1.region_admin = '09')  
      AND (sm1.forest_proc = '19' OR sm1.forest_admin = '19')
```

EXERCISE 11

Connect the columns you would use to join the three tables.

EXERCISE 12

Fill in the blanks to identify which tables you should use and to enable the table joins.

EXERCISE 13

Create a template for this query in PL/SQL Developer.

EXERCISE 14

Use that template to create a query that produces:

1. sm.1.project_name
2. sm1.setting_id
3. sm.measurement_date
4. sd.selection_method_type
5. sd.sample_expansion_factor
6. sc.selection_criteria_no
7. sc.subpop_code_value
8. sc.subpop
9. sc.subpop_min_value
10. sc.subpop_max_value

Where:

1. project_name = 'BROOKSTON'
2. setting_id = '091902007040003'

SUBQUERY: MOST RECENT MEASUREMENT

When a stand has been surveyed more than once over the years, it is often desirable to pull up only the latest measurement. This subquery is designed to do exactly that.

The test to see if it's needed

```
SELECT sm1,setting_id,
       COUNT(1) no_of_measurements
FROM   nrv_setting_measurements sm1
WHERE  <all the normal stuff>
GROUP BY sm1.setting_id
HAVING COUNT(1) > 1
```

Concept

```
SELECT <expressions>
  FROM nrv_setting_measurements sm1
  WHERE <all the normal stuff>
    AND sm1.setting_id
  HAVING COUNT(1) > 1
```

The query in full

```
SELECT <expressions>
  FROM nrv_setting_measurements sm1
  WHERE <all the normal stuff>
    AND sm1.measurement_date =
      (SELECT MAX(sm1_i.measurement_date)
       FROM nrv_setting_measurements sm1_i
       WHERE sm1_i.setmeas_cn_of IS NULL
         AND sm1_i.setting_id = sm1.setting_id
       GROUP BY sm1_i.setting_id)
  SELECT sm1.project_name,
    sm1.setting_id,
    sm1.measurement_date,
    sm2.level_2_id plot,
    tm.live_dead,
    tm.species_code,
    tm.diameter,
    tm.height
  FROM nrv_setting_measurements sm1,
    nrv_setting_measurements sm2,
    nrv_tree_measurements tm
  WHERE sm1.setmeas_cn_of IS NULL
    AND sm1.cn = sm2.setmeas_cn_of
    AND sm2.cm = tm.setmeas_cn
    AND (sm1.region_proc = '09' OR sm1.region_admin = '09')
    AND (sm1.forest_proc = '19' OR sm1.forest_admin = '19')
    AND sm1.setting_id = '091902007040003'
    AND _____ =
      (SELECT _____
       FROM _____
       WHERE _____
        AND _____
       GROUP BY _____)
```

EXERCISE 15

Modify the query you produced for [Exercise 10](#) to select:

1. Any project_name (remove the "AND sm1.project_name = 'BROOKSTON' * requirement). Run that before making any other changes to see what you'd get.
2. Only the latest measurement (i.e., use the most recent measurement subquery by filling it in in the blank lines below. Run it and compare the results with what you got above.

EXERCISE ANSWERS

EXERCISE 1

Review the instructions in this document on [creating filters](#).

EXERCISE 2

```
SELECT project name,
       setting_id,
       measurement_date
FROM   nrv_setting_measurements
WHERE  setmeas_cn_of IS NULL
       AND region_proc = '09'
       AND forest_proc = '19'
       AND level_1_alias = 'STAND'
```

EXERCISE 3

It depends. This number will change over time as exams are conducted on this forest.

EXERCISE 4

```
SELECT project name,
       setting_id,
       measurement_date
FROM   nrv_setting_measurements
WHERE  setmeas_cn_of IS NULL
       AND region_proc = '09'
       AND forest_proc = '19'
       AND level_1_alias = 'STAND'
ORDER BY project_name,
       setting_id
```

EXERCISE 5

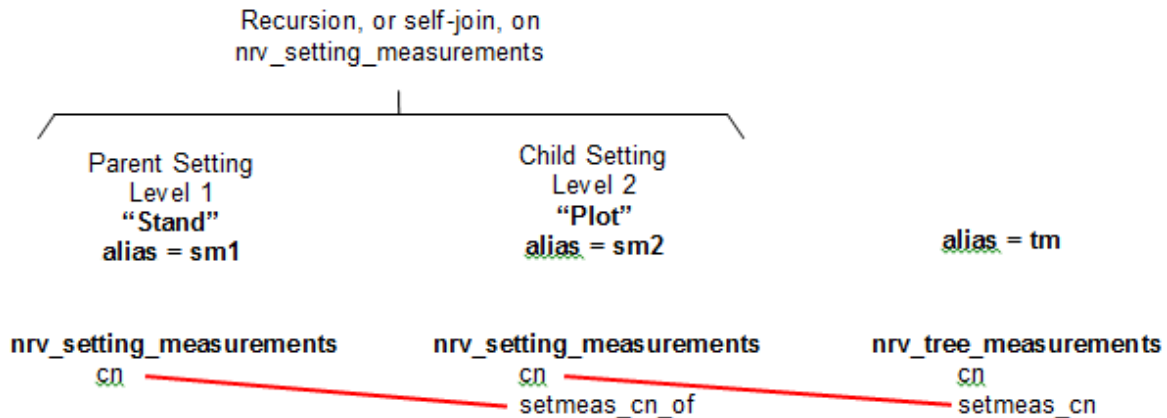
```
SELECT project name,
       setting_id,
       measurement_date,
       COUNT(1),
       SUM(setting_size)
FROM   nrv_setting_measurements
WHERE  setmeas_cn_of IS NULL
       AND region_proc = '09'
       AND forest_proc = '19'
       AND level_1_alias = 'STAND'
GROUP BY project_name,
       setting_id,
       measurement_date
ORDER BY SUM(setting_size)
```

EXERCISE 6

After running the query for Exercise 5, use “Copy results to Excel” hint earlier in this document to generate an Excel spreadsheet. Use Excel's Data/Sort to re-organize the data if you want, then use the Chart Wizard to produce a quick graph.

EXERCISE 7

Figure 30: Recursion, or self-join, on nrv_setting_measurements



EXERCISE 8

```
SELECT <column 1>
      ...
      <column n>
FROM   nrv_setting_measurements sm1,
       nrv_setting_measurements sm2,
       nrv_tree_measurements tm
WHERE  setmeas_cn_of IS NULL
      AND sm1.cn = sm2.setmeas_cn_of
      AND sm2.cn = tm.setmeas_cn
      ...
      AND (sm1.region_proc = '09' OR sm1.region_admin = '09')
      AND (sm1.forest_proc = '19' OR sm1.forest_admin = '19')
```

EXERCISE 9

Review the instructions on pages 9-11 of this document on how to **create a template**.

EXERCISE 10

Review the instructions on page 12 of this document on **using a template to create a query**.

```
SELECT sm1.project_name,
       sm1.setting_id,
       sm1.measurement_date,
       sm2.level_2_id,
       tm.live_dead,
       tm.species_symbol,
```

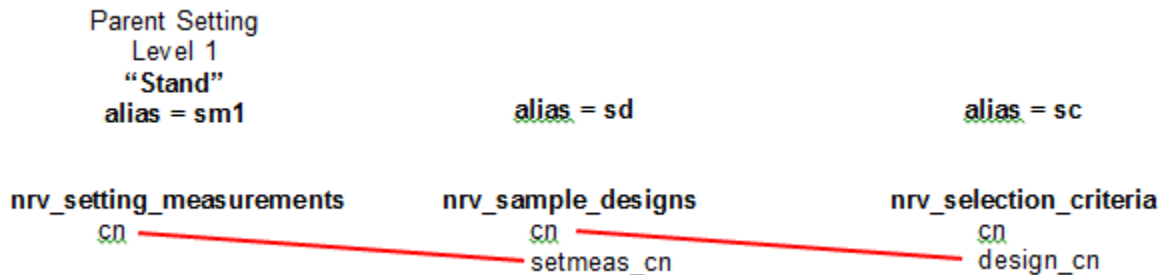
```

        tm.diameter,
        tm.height
    FROM nrv_setting_measurements sm1,
         nrv_setting_measurements sm2,
         nrv_tree_measurements tm
    WHERE sm1.setmeas_cn_of IS NULL
          AND sm1.cn = sm2.setmeas_cn_of
          AND sm2.cn = tm.setmeas_cn
          AND sm1.project_name = 'BROOKSTON'
          AND sm1.setting_id = '091902007040003'
          AND (sm1.region_proc = '09' OR sm1.region_admin = '09')
          AND (sm1.forest_proc = '19' OR sm1.forest_admin = '19')

```

EXERCISE 11

Figure 31: Flow of parent setting



EXERCISE 12

```

SELECT <column 1>,
      ...
      <column n>
    FROM nrv_setting_measurements sm1,
         nrv_sample_designs sd,
         nrv_selection_criteria sc
    WHERE sm1.setmeas_cn_of IS NULL
          AND sm1.cn = sd.setmeas_cn
          AND sd.cn = sc.setmeas_cn
          ...
          AND (sm1.region_proc = '08' OR sm1.region_admin = '08')
          AND (sm1.forest_proc = '19' OR sm1.forest_admin = '19')

```

EXERCISE 13

Review the instructions on pages 9-11 of this document on [how to create a template](#).

EXERCISE 14

```

SELECT sm1.project_name,
       sm1.setting_id,
       sm1.measurement_date,
       sd.selection_method_type,
       sd.sample_expansion_factor,
       sc.selection_criteria_no,
       sc.subpop_code_value,

```

```

        sc.subpop,
        sc.subpop_min_value,
        sc.subpop_max_value
FROM nrv_setting_measurements sm1,
     nrv_sample_designs sd,
     nrv_selection_criteria sc
WHERE sm1.setmeas_cn_of IS NULL
      AND sm1.cn = sd.setmeas_cn
      AND sd.cn = sc.design_cn
      AND sm1.level_1_alias = 'STAND'
      AND sm1.project_name = 'BROOKSTON'
      AND sm1.setting_id = '091902007040003'
      AND (sm1.region_proc = '09' OR sm1.region_admin = '09')
      AND (sm1.forest_proc = '19' OR sm1.forest_admin = '19')

```

EXERCISE 15

```

SELECT sm1.project_name,
       sm1.setting_id,
       sm1.measurement_date,
       sm2.level_2_id,
FROM   nrv_setting_measurements sm1,
       nrv_setting_measurements sm2,
       nrv_tree_measurements tm
WHERE  sm1.setmeas_cn_of IS NULL
      AND sm1.cn = sm2.setmeas_cn_of
      AND sm2.cn = tm.setmeas_cn
      AND sm1.level_1_alias = 'STAND'
      --AND sm1.project_name = 'BROOKSTON'
      AND sm1.setting_id = '091902007040003'
      AND (sm1.region_proc = '09' OR sm1.region_admin = '09')
      AND (sm1.forest_proc = '19' OR sm1.forest_admin = '19')
      AND sm1.measurement_date =
        (SELECT MAX(sm1_i.measurement_date)
         FROM nrv_setting_measurements sm1_i
         WHERE sm1_i.setmeas_cn_of IS NULL
              AND sm1_i.setting_id = sm1.setting_id
         GROUP BY sm1_i.setting_id)

```

PART 5: EXAMPLES FROM THE REAL WORLD

SINGLE-TABLE QUERIES

1. Common uses
 - a. Single-level (e.g., stand) observations
 - b. Counting records in a table
 - i. Settings
 - ii. Plots
 - iii. Trees

2. Column aliases
 - a. Mostly for clearer reporting
 - b. One string
3. COUNT and SUM – used with GROUP BY
4. IN – used to identify where something is in a list

QUERY 01 – NUMBER OF STANDS AND SUM OF ACRES BY PROJECT NAME

Where:

1. region_admin = '08'
- ```

SELECT project name
 COUNT(1) no_of_stands,
 SUM(setting_size) acres
FROM nrv_setting_measurements
WHERE setmeas_cn_of IS NULL
 AND archive_flag IS NULL
 AND (sm1.region_proc = '08' OR sm1.region_admin = '08')
GROUP BY project_name
ORDER BY project_name

```

Figure 32: Results of Query 01

|    | PROJECT_NAME       | NO_OF_STANDS | ACRES |
|----|--------------------|--------------|-------|
| 1  | 5555               | 1            | 1008  |
| 2  | ABCD               | 2            |       |
| 3  | ATT_RCW            | 1            | 16    |
| 4  | BEYER              | 1            |       |
| 5  | BIO MASS TEST      | 1            | 5     |
| 6  | BRANDYWINE TERRACE | 3            |       |
| 7  | BUFFMAN            | 1            |       |
| 8  | BWT                | 4            | 162   |
| 9  | CISC_LEGACY_DATA   | 1            | 68    |
| 10 | CNK C12            | 23           | 720   |
| 11 | CNK C2             | 35           | 914   |
| 12 | DRYMAN             | 1            |       |
| 13 | FMC                | 112          | 2529  |
| 14 | HWY RCW            | 1            | 25    |

## QUERY 02 – NUMBER OF STANDS AND SUM OF ACRES BY PROJECT NAME AND EV\_CODE

Modify [Query 01](#) above so that:

1. ev\_code is also selected
2. region\_admin = '09'
3. archive\_flag IS NULL

```

SELECT project_name,
 Ev_code,

```



```

COUNT(1) no_of_stands,
SUM(setting_size) acres
FROM nrv_setting_measurements
WHERE setmeas_cn_of IS NULL
AND archive_flag IS NULL
AND (sm1.region_proc = '09' OR sm1.region_admin = '09')
GROUP BY project_name,
ev_code
ORDER BY project_name,
ev_code

```

Figure 33: Results of Query 02

|    | PROJECT_NAME | EV_CODE | NO_OF_STANDS | ACRES |
|----|--------------|---------|--------------|-------|
| 1  | 2003 CSE     | 32      | 4            | 27    |
| 2  | 2003 CSE     | 35      | 1            | 1     |
| 3  | 2003 CSE     | 42      | 1            | 9     |
| 4  | 2003 CSE     | 51      | 3            | 26    |
| 5  | 2003 CSE     | 53      | 21           | 283   |
| 6  | 2003 CSE     | 54      | 10           | 92    |
| 7  | 2003 CSE     | 55      | 1            | 9     |
| 8  | 2003 CSE     | 59      | 46           | 729   |
| 9  | 2003 CSE     | 75      | 1            | 22    |
| 10 | 2003 CSE     | 78      | 1            | 2     |
| 11 | 2003 CSE     | 99      | 2            | 29    |
| 12 | ABCD         | 2       | 1            |       |
| 13 | ABCD         |         | 2            |       |
| 14 | EDER         |         | 1            |       |
| 15 | NATIELLA     |         | 1            |       |
| 16 | VIITALA      |         | 1            |       |

## MULTI-TABLE QUERIES

1. Joining tables
  - a. Identify which tables are needed (FROM clause)
  - b. Figure out how the records in the tables are related to each other – identify the one or more columns in each table that are needed to join to the other tables (WHERE clause)
2. In most cases, use the primary and foreign keys (cn values) to join two or more related tables together
  - a. It beats using multiple columns, like in the old days
  - b. Pitfall: Cartesian joins (symptoms: too long, too many rows)
  - c. Pitfall: Not getting all of the rows you ought to (fix: outer join)
  - d. Exception: Joining to reference tables (no cn)
3. Common Uses of Multi-Table Queries
  - a. Join to a reference table (to get the whole name of something)

- b. Stand and plot (see Queries [03](#), [04](#), [05](#), and [06](#))
- c. Stand, plot, and tree (see Queries [07](#), [08](#), and [09](#))
- d. Stand, sample design, and selection criteria (see Query [10](#))

## QUERY 03 – JOIN TABLES

Join the `nrv_setting_measurements` and the `nrv_ev_cover_types` table to obtain the number of stands and the sum of the acres by region, forest, and ev code.

Where:

- 1. `archive_flag` IS NULL
- 2. `region_admin` = '09'

Modify [Query 02](#), and use the following columns to join the two tables:

Table 4: FSveg Roles, Descriptions, Permissions

| <code>nrv_setting_measurements</code> | <code>nrv_ev_cover_types</code> |
|---------------------------------------|---------------------------------|
| <code>ev_code</code>                  | <code>ev_code</code>            |
| <code>ev_code_ref</code>              | <code>ref_code</code>           |

Make sure to use an outer join (+) to pick up all setting records, not just the ones that have an `ev_code` that's associated with an `nrv_ev_cover_types` record. See what happens if you leave the outer joins off.

```

SELECT sm1.project_name,
 sm1.ev_code,
 ev.common_name,
 COUNT(1) no_of_stands,
 SUM(sm1.setting_size) acres
FROM nrv_setting_measurements sm1,
 nrv_ev_cover_types ev
WHERE sm1.setmeas_cn_of IS NULL
 AND sm1.archive_flag IS NULL
 AND (sm1.region_proc = '09' OR sm1.region_admin = '09')
 AND sm1.ev_code = ev.ev_code(+)
 AND sm1.ev_ref_code = ev.ref_code(+)
GROUP BY sm1.project_name,
 sm1.ev_code
 ev.common_name
ORDER BY sm1.project_name,
 sm1.ev_code,
 ev.common_name

```

Figure 34: Results of Query 03

|    | PROJECT_NAME | EV_CODE | COMMON_NAME                   | NO_OF_STANDS | ACRES |
|----|--------------|---------|-------------------------------|--------------|-------|
| 1  | 2003 CSE     | 32      | Shortleaf pine                | 4            | 27    |
| 2  | 2003 CSE     | 35      | Eastern redcedar              | 1            | 1     |
| 3  | 2003 CSE     | 42      | Eastern redcedar-hardwood     | 1            | 9     |
| 4  | 2003 CSE     | 51      | Post oak/blackjack oak        | 3            | 26    |
| 5  | 2003 CSE     | 53      | Black oak/scarlet oak/hickory | 21           | 283   |
| 6  | 2003 CSE     | 54      | White oak                     | 10           | 92    |
| 7  | 2003 CSE     | 55      | Northern red oak              | 1            | 9     |
| 8  | 2003 CSE     | 59      | Mixed oaks                    | 46           | 729   |
| 9  | 2003 CSE     | 75      | Sycamore-pecan/American elm   | 1            | 22    |
| 10 | 2003 CSE     | 78      | Black walnut                  | 1            | 2     |
| 11 | 2003 CSE     | 99      | Open                          | 2            | 29    |
| 12 | ABCD         | 2       | Red pine                      | 1            |       |
| 13 | ABCD         |         |                               | 2            |       |
| 14 | EDER         |         |                               | 1            |       |
| 15 | NATIELLA     |         |                               | 1            |       |
| 16 | VIITALA      |         |                               | 1            |       |

## QUERY 04 – CARTESIAN JOIN

Turn [Query 03](#) into a Cartesian join by commenting out the appropriate WHERE clause items.

## VIEWS

---

## THINGS TO KEEP IN MIND

---

1. Just because the computer generates it doesn't mean it's right. Always conduct an independent check – pull up a subset of your data and crunch the numbers in Excel or on a calculator. Do the results match those from your query?
2. If you don't have the time to do that, at least look over the results and see if they're in the ballpark.
3. There's usually more than one way to write an SQL query to get the data you need. If one method isn't proving very productive, look for ways of approaching your issue from a different angle.
4. Keep examples of really good queries – you can modify them in the future to answer similar sorts of questions.
5. Remember, you're not in this alone, nor should it be an exercise in excruciating pain. Call your Regional FSVeg Representative or Roy Mita (tel: 970.295.5760) if you find yourself spinning your wheels.