United States Department of Agriculture

US Forest Service  
Natural Resource Manager (NRM)

# FSVeg Common Stand Exam – Querying FSVeg

June 2015

Table of Contents

[Overview 4](#_Toc421011386)

[Part 1: PL/SQL Developer & Command Line Authenticator 4](#_Toc421011387)

[Filtering Objects 9](#_Toc421011388)

[Built-In Aids 13](#_Toc421011389)

[Templates 15](#_Toc421011390)

[Using a Template 18](#_Toc421011391)

[Exercise 1 19](#_Toc421011392)

[To Run a Query 19](#_Toc421011393)

[Part 2: Basic SQL Query Format 19](#_Toc421011394)

[Exercise 2 19](#_Toc421011395)

[Exercise 3 20](#_Toc421011396)

[Exercise 4 20](#_Toc421011397)

[Use of Count and Sum with Group By 20](#_Toc421011398)

[Exercise 5 20](#_Toc421011399)

[Exercise 6 20](#_Toc421011400)

[Part 3: FSVeg Table Structure 20](#_Toc421011401)

[Part 4: Developing Queries 24](#_Toc421011402)

[Retrieving Stand, Plot, and Tree Data Tables and Columns 24](#_Toc421011403)

[Exercise 7 25](#_Toc421011404)

[Exercise 8 25](#_Toc421011405)

[Exercise 9 25](#_Toc421011406)

[Exercise 10 25](#_Toc421011407)

[Retrieving Stand, Sample Designs, and Selection Criteria Tables and Columns 25](#_Toc421011408)

[Exercise 11 25](#_Toc421011409)

[Exercise 12 26](#_Toc421011410)

[Exercise 13 26](#_Toc421011411)

[Exercise 14 26](#_Toc421011412)

[Subquery: Most Recent Measurement 26](#_Toc421011413)

[Exercise 15 27](#_Toc421011417)

[Exercise Answers 28](#_Toc421011418)

[Exercise 1 28](#_Toc421011419)

[Exercise 2 28](#_Toc421011420)

[Exercise 3 28](#_Toc421011421)

[Exercise 4 28](#_Toc421011422)

[Exercise 5 28](#_Toc421011423)

[Exercise 6 29](#_Toc421011424)

[Exercise 7 29](#_Toc421011425)

[Exercise 8 29](#_Toc421011426)

[Exercise 9 29](#_Toc421011427)

[Exercise 10 29](#_Toc421011428)

[Exercise 11 30](#_Toc421011429)

[Exercise 12 30](#_Toc421011430)

[Exercise 13 30](#_Toc421011431)

[Exercise 14 30](#_Toc421011432)

[Exercise 15 31](#_Toc421011433)

[Part 5: Examples From the Real World 31](#_Toc421011434)

[Single-Table Queries 31](#_Toc421011435)

[Query 01 – Number of stands and sum of acres by project name 32](#_Toc421011436)

[Query 02 – Number of Stands and Sum of Acres by Project Name and EV\_CODE 32](#_Toc421011437)

[Multi-Table Queries 33](#_Toc421011438)

[Query 03 – Join Tables 34](#_Toc421011439)

[Views 34](#_Toc421011440)

[Things to Keep in Mind 34](#_Toc421011441)

## Overview

1. Query Development Environment: PL/SQL Developer
2. Basic Query Format
3. FSVeg Table Structure: Primary Components
   1. Data: stand, plot, tree, etc.
   2. 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](http://fsweb.nris.fs.fed.us/For_Developers/plsqldev.shtml).

Figure 1: Get PL/SQL Developer

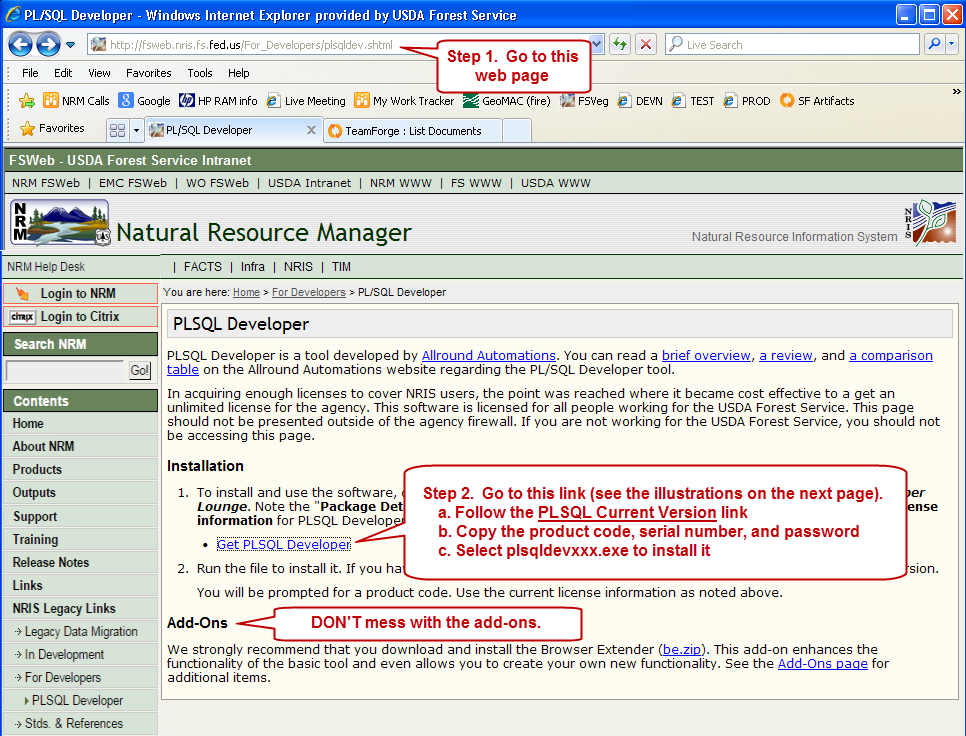


Figure 2: Link on TeamForge

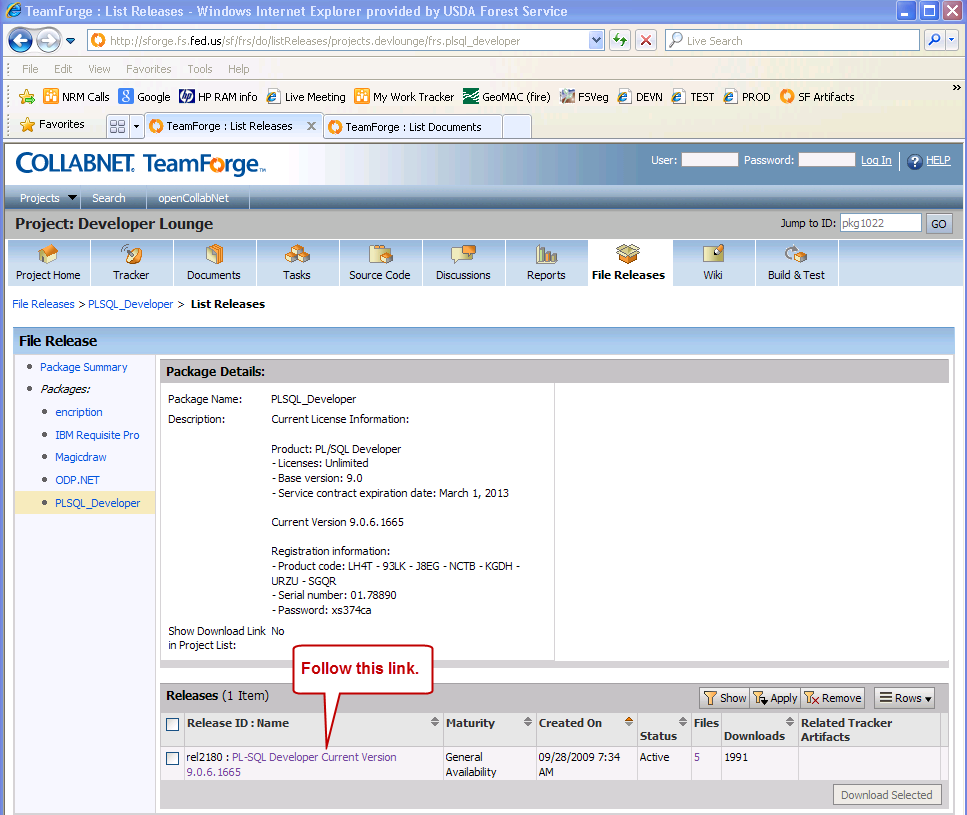
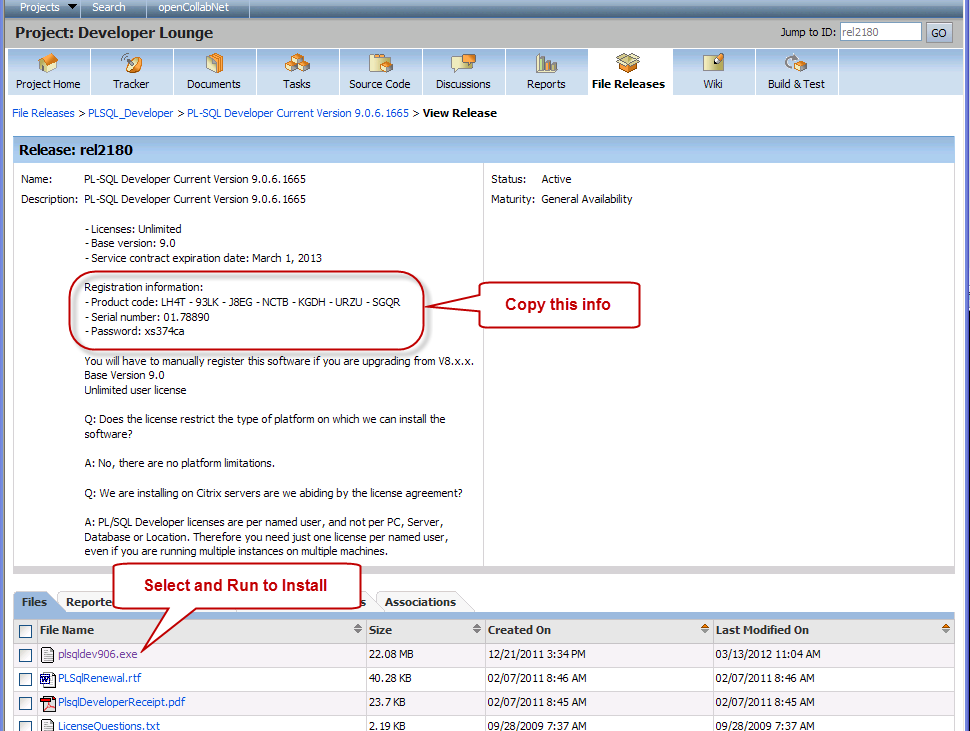


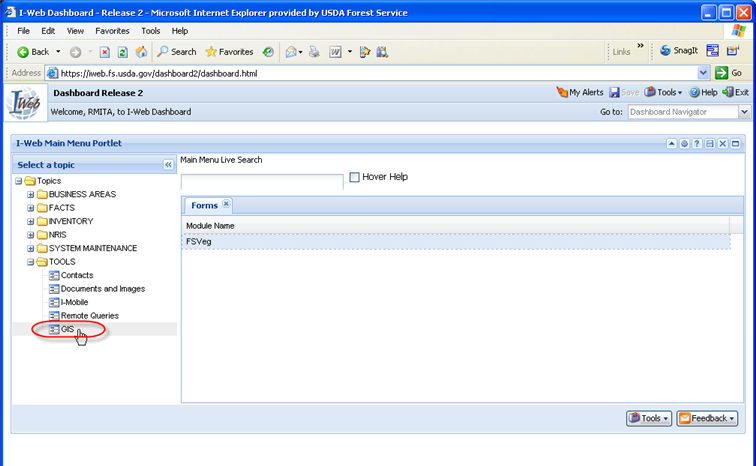
Figure 3: PL/SQL content to copy[[1]](#footnote-1)



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](https://iweb.fs.usda.gov/gisutils/cl_auth/cl_auth.html) 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:

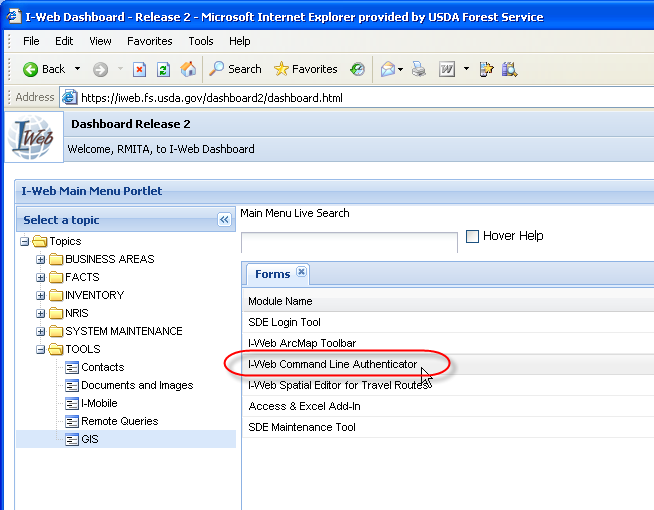
1. Go to the I-Web site
2. Select Default I-Web Dashboard
3. Go through the e-Authentication process; you will end up at the I-Web Dashboard, as seen in Figure 4.
4. Select, from the menu tree on the left side of the screen: **Tools / GIS**

Figure 4: I-Web Dashboard

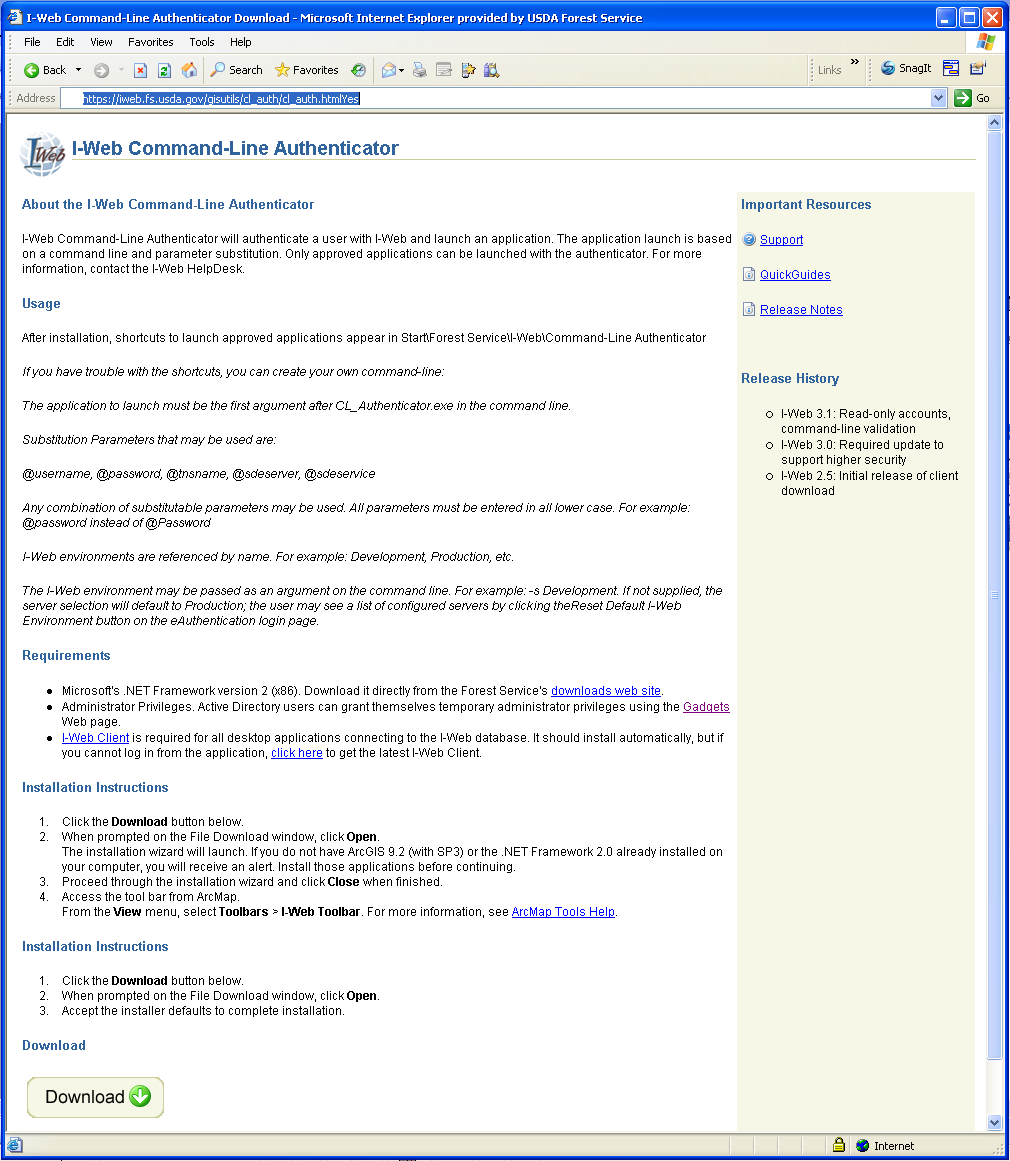


1. Select the I-Web Command Lind Authenticator

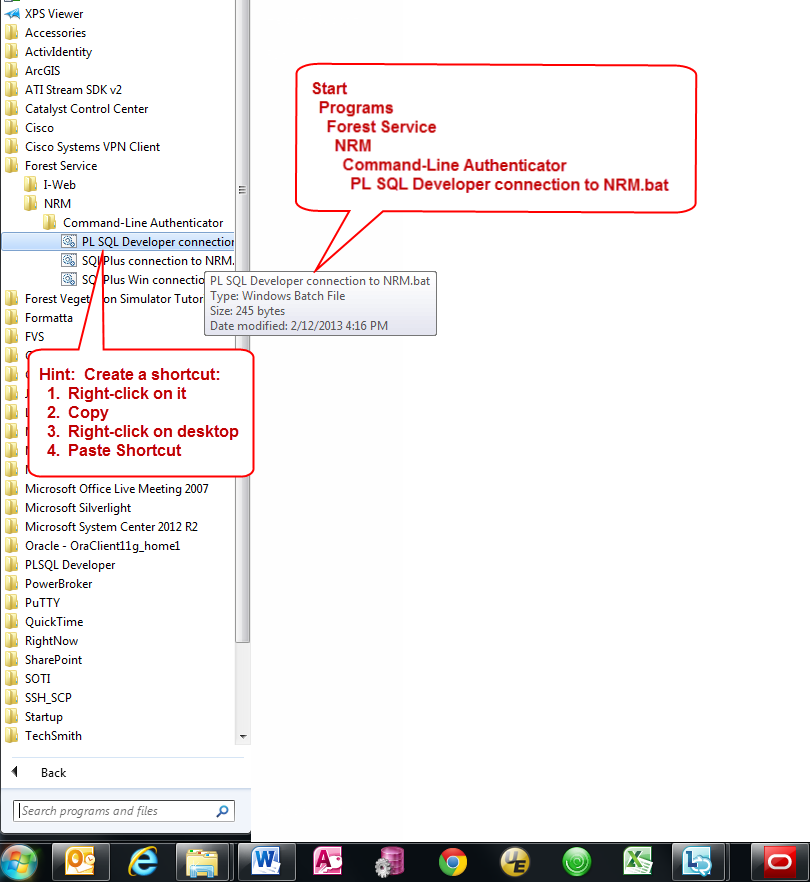
Figure 5: Access the I-Web Command Line Authenticator



1. 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 FSVeg data at NITC.

### Filtering Objects

Figure 6: Open the Filters

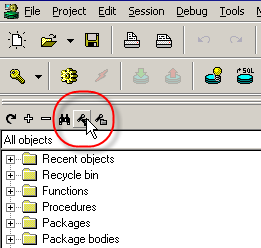


Figure 7: Create a New Filter

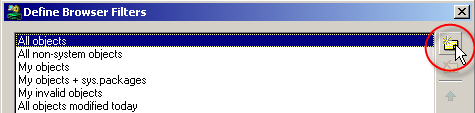
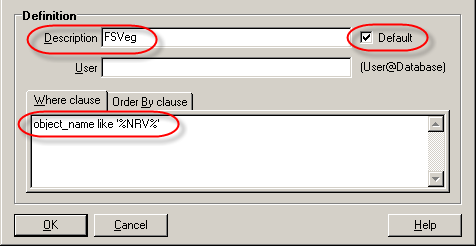


Table 1: FSVeg Roles, Descriptions, Permissions

| **Description** | **WHERE clause** |
| --- | --- |
| **FSVeg** | object\_name LIKE ‘%NRV%’ |
| **FSVeg Spatial** | bject\_name LIKE ‘%NRIS%APRV’ |

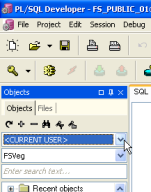
Figure 8: Set FSVeg or FSVeg Spatial as your Default Filter; click OK

****

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



1. 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
2. Select the desired Selected User from the list

Figure 10: Accessing FS\_NRIS\_FSVEG (Selected User) Object

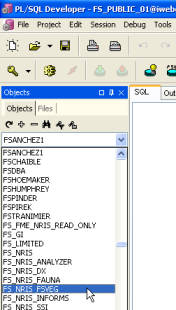
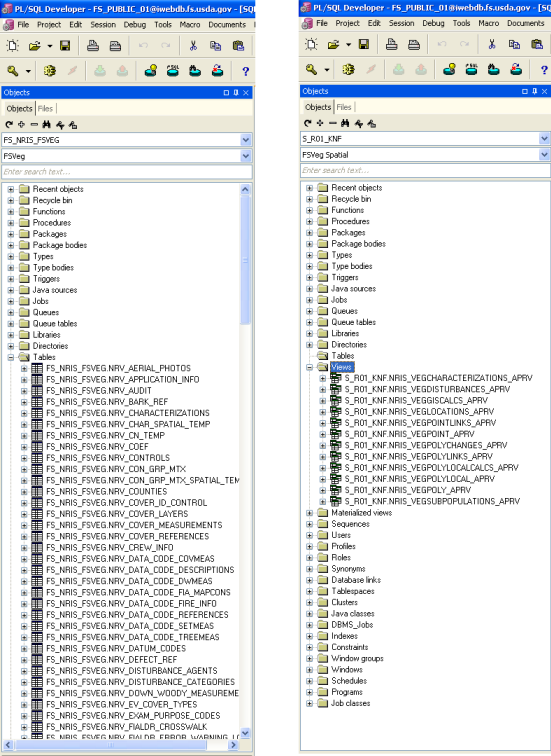


Table 2: Selected users by application

| **Application** | **Selected User** |
| --- | --- |
| **FSVeg** | FS\_NRIS\_FSVEG |
| **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

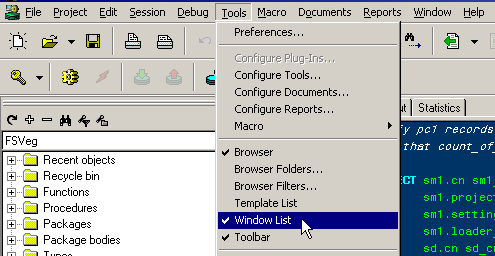


Figure 13: Columns for tables and views

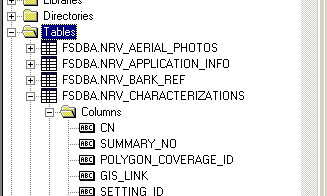


Figure 14: Table autofill

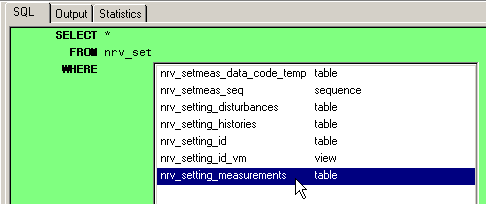


Figure 15: Column autofill (use table aliases)

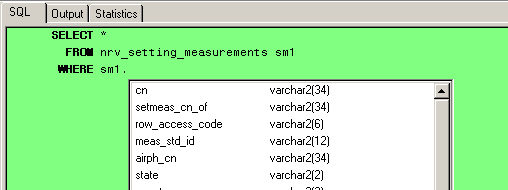


Figure 16: Obtain just a page of data, or all of it

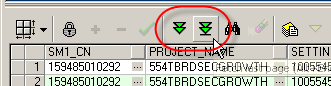
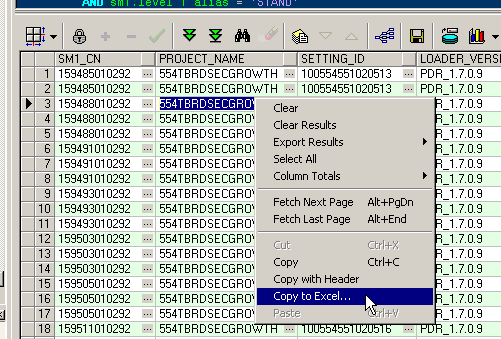


Figure 17: Copy results to Excel



### Templates

Figure 18: Showing the Templace List

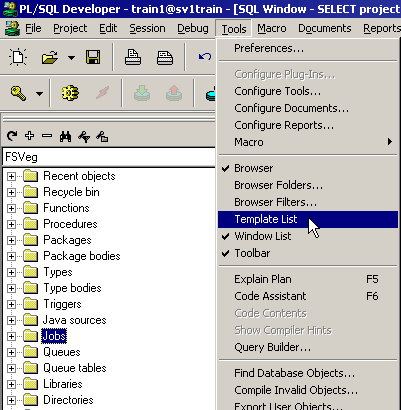


Figure 19: Template list

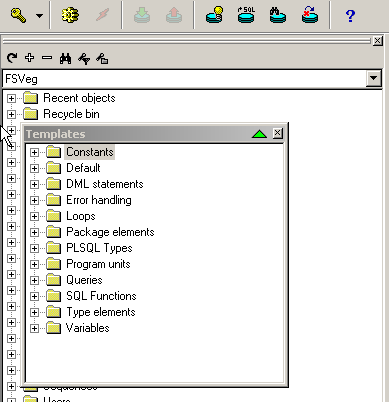


Figure 20: Move Template window

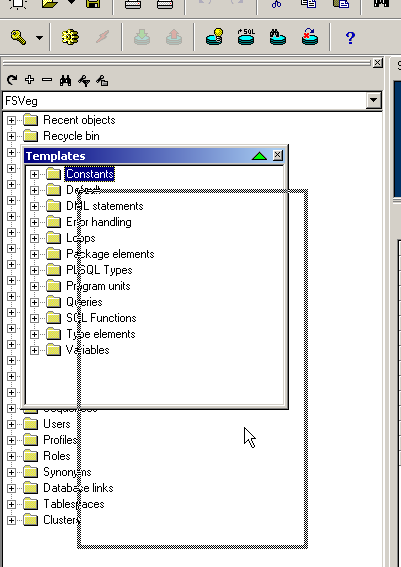


Figure 21: Template lists repositioned as a tab

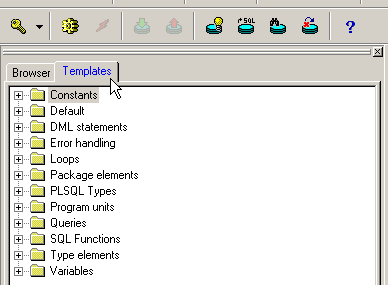
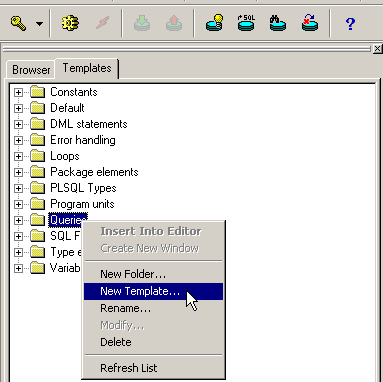


Figure 22: Create a new template



(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

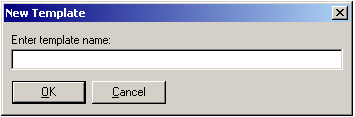
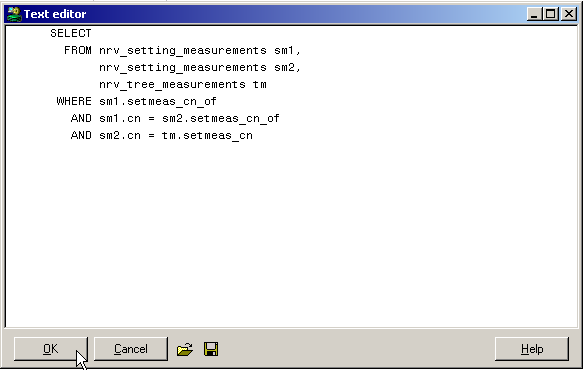


Figure 24: Text editor window



### Using a Template

Figure 25: Step 1 – Open an SQL window

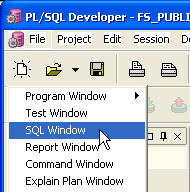
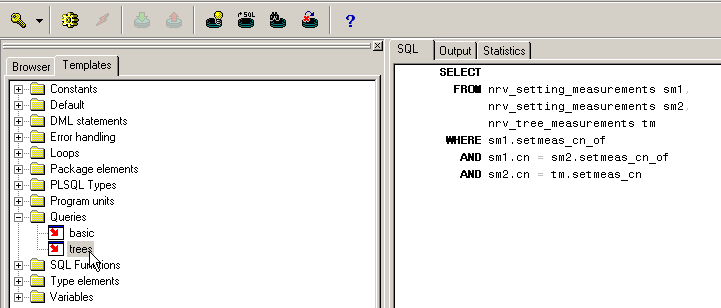


Figure 26: Step 2 – Navigate to and Select Template



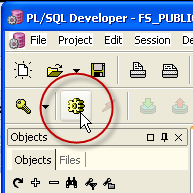
### Exercise 1

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



1. 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
   1. Caveats
      1. Not an entity relationship diagram (ERD)
      2. Not complete at all
      3. Not for surveys that have more than 2 levels
   2. How to use it
      1. Depending on your query strategy
         1. Which tables will you go after
         2. Order of joins (smallest number of rows first – a production issue)
      2. 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
      3. Start at an “anchor” table, then work your way around
2. CNs
   1. VARCHAR2(34)
   2. A pain at first, but they will save your sanity in the long run
3. Audit columns
   1. created\_by, created\_date\_in\_instance
   2. modified\_by, modified\_date\_in\_instance (last update only)
4. Collector version, loader version
   1. Collector: paper forms or PDR version x?
   2. Loader: PDR loader, data entry forms, legacy loader?
   3. 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’ |

1. nrv\_setting\_measurements: recursion; has both parent and child records
   1. Parent setting (“stand”): setmeas\_cn\_of = NULL
   2. Child setting (“plot”): setmeas\_cn\_of = parent setting cu
2. archive\_flat in nrv\_setting\_measurements:
   1. A survey is archived when:
      1. Superseded by a newer survey
      2. Altered stand conditions render existing survey no longer relevant
      3. Stand redelineation
   2. ‘Y’ denotes setting records that are no longer considered current
   3. 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\_setting\_measurements.

*c:\my documents\rel\_map\_2.ppt 03-17-2004*

nrv\_cover\_measurements

cn

setmeas\_cn

selcrit\_cn

nrv\_ down\_woody

cn

setmeas\_cn

selcrit\_cn

nrv\_tree\_measurements

cn

setmeas\_cn

selcrit\_cn

nrv\_sample\_designs

cn

setmeas\_cn

nrv\_setting\_measurements

cn

meas\_std\_id

nrv\_plot\_counts

cn

design\_cn

setmeas\_cn

nrv\_tree\_disturbances

cn

tremeas\_cn

nrv\_selection\_criteria

cn

design\_cn

nrv\_aerial\_photos

cn

setmeas\_cn

nrv\_setting\_histories

cn

setmeas\_cn

nrv\_setting\_disturbances

cn

setmeas\_cn

nrv\_ setting\_measurements

cn

setmeas\_cn\_of

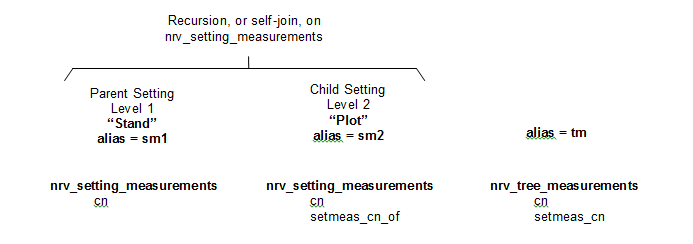
nrv\_measurement\_standards

id

## 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** nrv\_setting\_measurements sm1,[[2]](#footnote-2)

nrv\_setting\_measurements sm2,[[3]](#footnote-3)

nrv\_tree\_measurements tm[[4]](#footnote-4)

**WHERE** sm1.setmeas\_cn\_of IS NULL

AND \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_[[5]](#footnote-5)

AND \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_[[6]](#footnote-6)

...

AND (sm1.region\_proc = ‘09’ OR sm1.region\_admin = ‘09’)[[7]](#footnote-7)

AND (sm1.forest\_proc = ‘19’ OR sm1.forest\_admin = ‘19’)

### 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

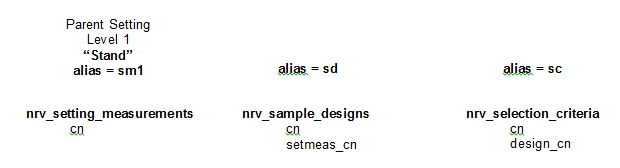
### Exercise 10

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](#_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

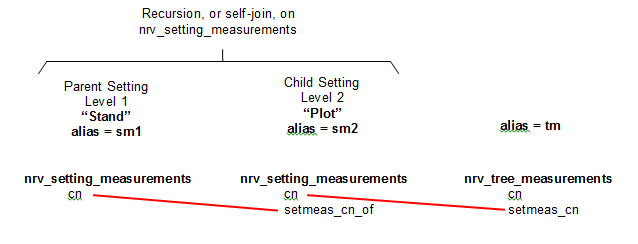
ORDER BY SUM(setting\_size)

### 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
   1. Single-level (e.g., stand) observations
   2. Counting records in a table
      1. Settings
      2. Plots
      3. Trees
2. Column aliases
   1. Mostly for clearer reporting
   2. 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

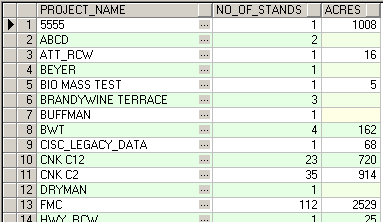
AND archive\_flag IS NULL

AND (sm1.region\_proc = ‘08’ OR sm1.region\_admin = ‘08’)

GROUP BY project\_name

ORDER BY project\_name

Figure 32: Results of Query 01



### Query 02 – Number of Stands and Sum of Acres by Project Name and EV\_CODE

Modify [Query 01](#_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’)

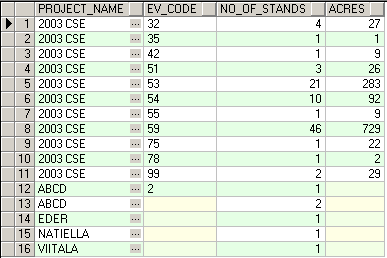
GROUP BY project\_name,

ev\_code

ORDER BY project\_name,

ev\_code

Figure 32: Results of Query 02



### Multi-Table Queries

1. Joining tables
   1. Identify which tables are needed (FROM clause)
   2. 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
   1. It beats using multiple columns, like in the old days
   2. Pitfall: Cartesian joins (symptoms: too long, too many rows)
   3. Pitfall: Not getting all of the rows you ought to (fix: outer join)
   4. Exception: Joining to reference tables (no cn)
3. Common Uses of Multi-Table Queries
   1. Join to a reference table (to get the whole name of something)
   2. Stand and plot (see Queries [03](#_Query_03_–), 04, 05, and 06)
   3. Stand, plot, and tree (see Queries 07, 08, and 09)
   4. 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](#_Query_02_–), and use the following columns to join the two tables:

Table 4: FSVeg Roles, Descriptions, Permissions

| **nrv\_setting\_measurements** | **nrv\_ev\_cover\_types** |
| --- | --- |
| ev\_code | ev\_code |
| ev\_code\_ref | ref\_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

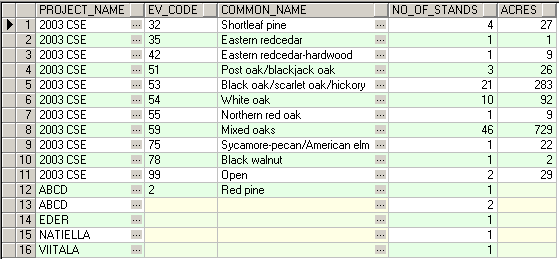
ev.common\_name

ORDER BY sm1.project\_name,

sm1.ev\_code,

ev.common\_name

Figure 32: Results of Query 03



### Query 04 – Cartesian Join

Turn [Query 03](#_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.

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. [↑](#footnote-ref-1)
2. “Stand” [↑](#footnote-ref-2)
3. “Plot” [↑](#footnote-ref-3)
4. “Tree” [↑](#footnote-ref-4)
5. Stand-to-Plot join [↑](#footnote-ref-5)
6. Plot-to-Tree join [↑](#footnote-ref-6)
7. Make sure you narrow down your search with something like these two lines, since the entire nation’s data is at NITC [↑](#footnote-ref-7)