GRIN-Global Library

Revision Date
July 8, 2014

Contents

Dynamic Folder Examples **2**
Accession (Single) 2
Accession (Range) 2
Accessions (Several) 2
Species 2
Collected –in a specific country 2
Origin (Source) 3
Orders (Pending) 3
Orders (Filled) 3
Traits (Sorghum - KERCOLOR 2 4) 3
Inventory (Available) 3
Inventory (Low) 3
Methods 4

SQL Examples **5**
List the Top Accession Names for your Site 5
Find Non-Exists 6
Display Field Names 6
Display Codes and Code Groups 7
Count of Items Shipped by State 7
Query for Crops with Observations at a Site 8
List Accession_Inventory Group Names 8
List the Unique Indexes and Their Fields 9

Reports **9**

Dynamic Folder Examples

Refer to the document online for details and tips about Dynamic Folders.

Note that these queries were used with data in a database with data copied from GRIN. They are here to for illustration, but will need to be modified to match your data.

Accession (Single)

@accession.accession_number_part1 = 'PI' AND
@accession.accession_number_part2 = 600000

Accession (Range)

(@accession.accession_number_part2 > 500000 AND @accession.accession_number_part2 < 500100)

Accessions (Several)

@accession.accession_number_part1 = IN ('PI' , 'CRIB') AND
@accession.accession_number_part2 > 500000

Species

@taxonomy_species.name = 'Ribes cereum'

Collected –in a specific country

@accession_source.source_type_code = 'COLLECTED' AND @accession_source.geography_id IN (926, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010, 1011, 1012, 1013, 1014, 1015, 1016, 1017, 1018, 1019, 1020, 1021, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051) AND @accession_source.is_origin = 'Y' AND @accession.taxonomy_species_id IN (27512, 27513, 27514)

same query, but written with more obvious parameters:
Born in the USA
@taxonomy_species.name = 'Helianthus tuberosus'
AND @accession_source.source_type_code = 'COLLECTED'
AND @geography.country_code = 'USA'

Origin (Source)

@accession.taxonomy_species_id IN (27923)
AND @accession_source.geography_id IN (1041)
AND @accession_source.is_origin = 'Y'

Orders (Pending)

@order_request.ordered_date LIKE '%2013%' AND
@site.site_short_name = 'S9' AND
@order_request.order_type_code = 'DI' AND
@order_request.completed_date IS NULL

Orders (Filled)

@order_request_action.action_name_code = 'ORDFILLED' AND
@order_request.ordered_date LIKE '%2013%' AND
@site.site_id IN (16)

Traits (Sorghum - KERCOLOR 2 4)

@crop.name = 'SORGHUM'
AND @crop_trait.coded_name = 'KERCOLOR'
AND @crop_trait_code.code IN ('2', '4')

Inventory (Available)

@taxonomy_genus.genus_name = 'Zea' AND
@taxonomy_species.species_name = 'mays' AND
@taxonomy_species.subspecies_name = 'mexicana' AND
@site.site_short_name = 'NC7' AND
@inventory.availability_status_code = 'LOW'

Inventory (Low)

@inventory_maint_policy.maintenance_name LIKE 'NC7-maize.pop%' AND
@inventory.is_distributable = 'Y' AND
@inventory.is_available = 'Y' AND @inventory.availability_status_code = 'AVAIL' AND
@inventory.quantity_on_hand < 1500

Methods

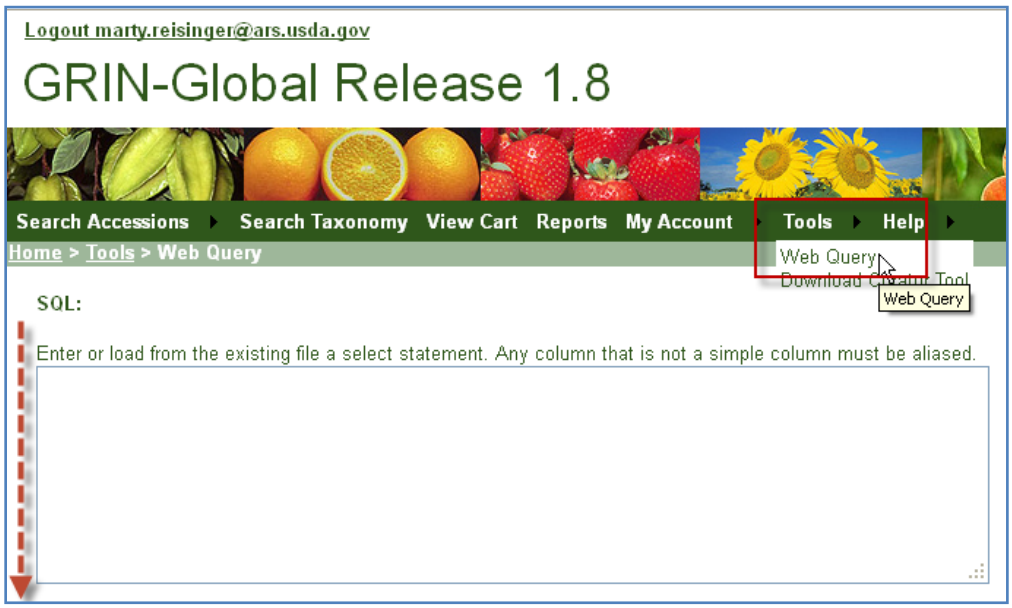
The following sample Dynamic Query criteria will produce a list of methods records.

```
@method.method_id > 0
AND
@site.site_short_name = 'W6'
AND
@method.name LIKE 'ALFALFA%'
```

SQL Examples

These SQL statements can be used on the Public website, assuming that the user has been set up by the GRIN-Global admin to have this capability. The GG Admin will map the user's PW account to his/her CT user account. (NPGS: email Quinn Sinnot, indicating your desire to have this capability; provide the PW User Name you are using.)

Login; then select **Tools | Web Query** Supply the SQL code either by copying and pasting from a file, or use the window's Load SQL function.



List the Top Accession Names for your Site

```
SELECT
  a.accession_number_part1,
  a.accession_number_part2,
  a.accession_number_part3,
  ain.category_code,
  ain.plant_name,
  ain.plant_name_rank
FROM accession a
INNER join cooperator ac ON a.owned_by = ac.cooperator_id AND ac.site_id = 29
INNER JOIN accession_inv_name ain ON ain.accession_inv_name_id =
  (SELECT TOP 1 accession_inv_name_id
   FROM accession_inv_name
   INNER JOIN inventory ON accession_inv_name.inventory_id = inventory.inventory_id
   WHERE inventory.accession_id = a.accession_id
   ORDER BY accession_inv_name.plant_name_rank
  )
```

Determine your site id by using the Site dataview in the CT. In the example above, **ac.site_id = 29** represents Potato Germplasm Introduction Station (NR6).

Find Non-Exists

```
SELECT accession_number_part1, accession_number_part2,accession_number_part3
FROM accession a
INNER JOIN taxonomy_species ts ON a.taxonomy_species_id = ts.taxonomy_species_id
WHERE ts.name like 'Cicer%'
AND NOT EXISTS (SELECT * FROM inventory i
   WHERE a.accession_id = i.accession_id
      AND i.is_distributable = 'Y')
```

Display Field Names

SQL displays the table and field names in the database as well as the field titles (column headings) and descriptions. Note that the sl2.sys_lang_id = 1 i the WHERE clause indicates the English language; substitute the respective value to display another language

```
USE gringlobal;
SELECT
        st.table_name,
        stf.field_name,
        sl2.title as language_title,
        stfl.title as field_title,
        stfl.description as field_description
FROM
        sys_table st
        inner join sys_table_field stf
                on st.sys_table_id = stf.sys_table_id
        left join sys_table_field_lang stfl
                on stf.sys_table_field_id = stfl.sys_table_field_id
        left join sys_lang sl2
                on stfl.sys_lang_id = sl2.sys_lang_id
        left join sys_table_lang stl
                on st.sys_table_id = stl.sys_table_id
        left join sys_lang sl1
                on stl.sys_lang_id = sl1.sys_lang_id
WHERE
sl2.sys_lang_id = 1
/* and
st.table_name = 'accession_inv_name'
*/
ORDER BY
        st.table_name, stf.field_ordinal

/* comments */
```

Display Codes and Code Groups

```
USE
  gringlobal;

SELECT
  cv.code_value_id as value_id,

  cv.group_name as group_name,
  cv.value as gg_value,
  cvl.title as cvl_title


FROM
  code_value cv

  inner join code_value_lang cvl
              on cv.code_value_id = cvl.code_value_id
   left join sys_lang sl2
                          on cvl.sys_lang_id = sl2.sys_lang_id
WHERE
  sl2.sys_lang_id = 1
/* and   group_name = 'inventory_action'  */
/* use and clause to search for a specific code */

ORDER BY
  group_name,
  gg_value
```

Count of Items Shipped by State

```
SELECT
  g.adm1 AS State,
  count(*) AS Count

FROM order_request_item ori

INNER JOIN order_request o ON ori.order_request_id = o.order_request_id
INNER JOIN cooperator own ON ori.owned_by = own.cooperator_id
INNER JOIN site on own.site_id = site.site_id
INNER JOIN cooperator c ON o.final_recipient_cooperator_id = c.cooperator_id
INNER JOIN geography g ON c.geography_id = g.geography_id

WHERE ori.status_date >= '1/1/2013' AND ori.status_date < '1/1/2014'
  AND site.site_short_name = 'W6'
```

```
  AND o.order_type_code = 'DI'
  And ori.status_code = 'Shipped'
  AND g.country_code = 'USA'

 GROUP BY g.adm1

ORDER BY g.adm1
```

Question:

How do you query for the crops at a site with total number of observations per crop? In GC I use the following SQL:

```
select crop,count(*) Total_obs
  from prod.crop,prod.dsc,prod.ob where
  crop.cropno = dsc.cropno and dsc.dno = ob.dno
  and site = 'S9'
group by crop;
```

Answer:

Query for Crops with Observations at a Site

```
SELECT crop.name AS Crop,
     Count(*) AS Total_obs
FROM   crop
     INNER JOIN crop_trait ct
          ON crop.crop_id = ct.crop_id
     INNER JOIN crop_trait_observation cto
          ON ct.crop_trait_id = cto.crop_trait_id
     INNER JOIN cooperator c
          ON ct.owned_by = c.cooperator_id
     INNER JOIN site s
          ON c.site_id = s.site_id
WHERE  s.site_short_name = 'S9'
GROUP  BY crop.name
ORDER  BY crop.name
```

List Accession_Inventory Group Names

```
SELECT  group_name
FROM   accession_inv_group
```

List the Unique Indexes and Their Fields

```
SELECT t.name as tablename, i.name as indexname, tc.name as fieldname
FROM sys.tables t
	INNER JOIN sys.indexes i ON t.object_id = i.object_id
	INNER JOIN sys.index_columns ic ON i.object_id = ic.object_id AND i.index_id = ic.index_id
	INNER JOIN sys.columns tc ON i.object_id = tc.object_id AND ic.column_id = tc.column_id
WHERE i.is_unique = 1
ORDER BY t.name, i.name, ic.index_column_id
```

Reports

[to be developed]

Review the Reports spreadsheet for a list of available Curator Tool and Public Website reports (updated 2013, Dec.)