Due Date: Saturday, September 28, 2013 11:00 PM
Points: 40 points max
Turn In: The script and spool files turned in via the assignment drop box

General Directions

Use the plants tables- as shown in the plants script and discussion for the midterm exam. These tables do not have very many rows. Since you will be writing queries on the midterm exam without having feedback from a rows display, this should be good practice in evaluating your query- not just the output.

Tasks

Task 01: Display the customer id and last name for any customer who does not have any orders.

Task 02: We want to find plants for which we have no orders records. Show the plant id and common name.

The next three tasks are similar in some ways but do not use the same logic. Read all three of these before you work on any of them. These tasks all have a result set that uses the sample display shown here.

```
+----------+------------------------+--------------------+
| plant_id | common_name            | Scientific_Name    |
+----------+------------------------+--------------------+
|      876 | Black Iris             | Iris chrysographes |
|      936 | Rabbit-Ear Iris        | NULL               |
|      889 | White Cemetery Iris    | Iris albicans      |
+----------+------------------------+--------------------+
```

Task 03: Display the plant id, common name and scientific name for plants that someone ordered in July or August of 2013.

Task 04: Display the plant id, common name and scientific name for plants that no one ordered in July or August of 2013. This would include plants that were never ordered.

Task 05: Display the plant id, common name and scientific name for plants that someone has ordered but no one ordered the plant in July or August of 2013.

Task 06: Declare a variable name that will be used to hold a string pattern that may be part of a plant common name. Assign the value 'Buttercup' to the variable. Use the variable to find all plants that include that pattern in their common name. For each of these plants, display the common name and the scientific name- which is the genus name concatenated with the species name. Some plants do not have a scientific name; in that case the query displays a null.
Then assign the value 'Thistle' to the same variable and run the same query again.
Then assign the value 'Lupine' to the same variable and run the same query again.
Do *not* use the Like operator for this- use one of the functions discussed in this unit.

Sample run

```
+--------------------+--------------------+
| common_name        | ScientificName     |
+--------------------+--------------------+
| Common Buttercup   | Ranunculus acris   |
| Buttercup Betty    | NULL               |
| Creeping Buttercup | Ranunculus repens  |
+--------------------+--------------------+
```

Task 07: We have noticed that we have some plants still on hand in the plants table that have been discontinued! We need to know the Inventory Cost (defined as list price times on_hand) for

each of the plants that have been discontinued. Display the inventory cost rounded to the nearest hundred.

Sample run

```
+----------+-----------------+---------+------------+---------------+
| plant_id | common_name     | on_hand | list_price | InventoryCost |
+----------+-----------------+---------+------------+---------------+
|     2000 | Fuggy moss      |     100 |      12.35 |          1200 |
|     2001 | Stink horn      |      10 |       0.25 |             0 |
+----------+-----------------+---------+------------+---------------+
```

Task 08: Use the Find_In_Set function to display the customer id, customer last name and order date for all orders but exclude the orders by customers in the Midwestern states.

The Midwestern states are defined as: Illinois, Indiana, Iowa, Kansas, Michigan, Minnesota, Missouri, Nebraska, North Dakota, Ohio, South Dakota and Wisconsin. (I am sure you can find the state abbreviations for these.

Task 09: Set up a variable and use the rand function to generate a random integer between 10 and 50 inclusive, assigning it to the variable. Use the variable to display the plant_id and list price for all plants that cost less than that random value. Run this pair of statements twice (generate a second number and rerun the query.)