



Database Assignment 01

Deadline: September 29th 11:59pm

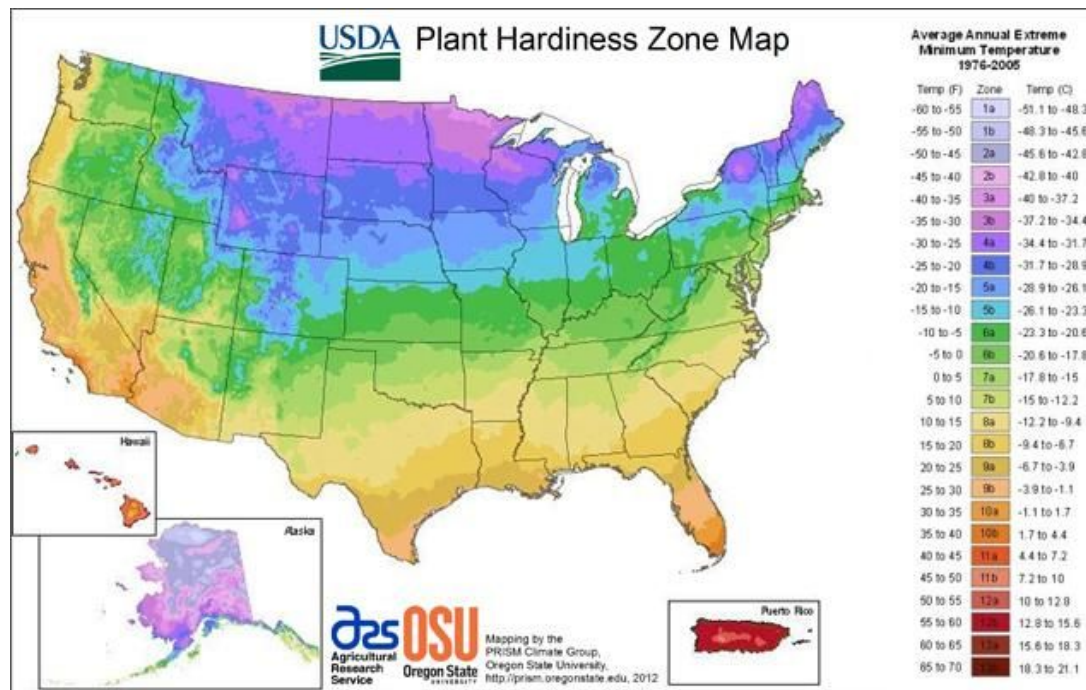
1. Introduction

The goal of this assignment is to have you create and query a database system of flowers. You are asked to submit a file named `flowers.sql` containing all the necessary SQL commands to create a database (named `flowers`), a few tables according to a predefined schema, populate those tables with some initial data, and query your database. Because this is a small database, you can use `INSERT INTO` syntax to populate your tables.

Your `flowers.sql` file MUST have a comment section in the beginning with the name(s) of the author(s) of the project. You are allowed to work together with another classmate. Teams of more than two students will NOT be accepted (NO exceptions). Only one of the members of the team needs to submit on Blackboard.

2. Zones

A flower vendor wants to market flowers that can be grown in a variety of zones. These zones define a range for the lowest (or highest) temperatures that the plants can accept during the year. See map below that illustrates the plant zones in the U.S. according to the United States Department of Agriculture (USDA).



Create a table named `Zones` with the attributes and assumptions indicated below.

- Attributes: the zone ID, the lowest and the highest accepted temperature.
- Assumptions: the ID will be the primary key and have one or two digits, the temperatures (in Fahrenheit) will be at most two digits and a possible minus sign, none of the temperatures can be `NULL`.

Populate table `Zones` so that it has the following rows:

id	lowerTemp	higherTemp
2	-50	-40
3	-40	-30
4	-30	-20
5	-20	-10
6	-10	0
7	0	10
8	10	20
9	20	30
10	30	40

3. Deliveries

The same flower vendor wants to use a code to explain the type of delivery for each flower. Create a table named `Deliveries` with the attributes and assumptions indicated below.

- Attributes: the delivery ID, the category or type of delivery, and the size of the delivery.
- Assumptions: the ID will be the primary key and have one one digit, the category will be at most five characters (pot, plant, hedge, shrub, tree), and the delivery size will be up to five digits with three decimal spaces (possibly `NULL`).

Populate table `Deliveries` so that it has the following rows:

id	categ	delSize
1	pot	1.500
2	pot	2.250
3	pot	2.625
4	pot	4.250
5	plant	NULL
6	bulb	NULL
7	hedge	18.000
8	shrub	24.000
9	tree	36.000

3. FlowersInfo

Create a table named `FlowersInfo` with the attributes and assumptions indicated below. Choose the most appropriate data types.

Attributes: an ID with three digits, common name, Latin name, the coolest and hottest zones where it can be grown, the delivery category, and the sun needs.

Assumptions: The ID will be the primary key, the attribute common name may have up to thirty characters, and the Latin name up to thirty-five characters. The attributes coolest zone, hottest zone, and delivery category will match the IDs from other tables, and the sun needs will be up to five characters, S for Sun, SH for Shade, P for Partial sun and any combination (StoP, StoSH, etc.). Your table definition should implement referential integrity whenever possible.

Populate table `FlowersInfo` so that it has the following rows:

id	comName	latName	cZone	hZone	deliver	sunNeeds
101	Lady Fern	Atbyrium filix-femina	2	9	5	SH
102	Pink Caladiums	C.x bortulanum	10	10	6	PtoSH
103	Lily-of-the-Valley	Convallaria majalis	2	8	5	PtoSH
105	Purple Liatris	Liatris spicata	3	9	6	StoP
106	Black Eyed Susan	Rudbeckia fulgida var. specios	4	10	2	StoP
107	Nikko Blue Hydrangea	Hydrangea macrophylla	5	9	4	StoSH
108	Variegated Weigela	W. florida Variegata	4	9	8	StoP
110	Lombardy Poplar	Populus nigra Italica	3	9	9	S
111	Purple Leaf Plum Hedge	Prunus x cistena	2	8	7	S
114	Thorndale Ivy	Hedera belix Thorndale	3	9	1	StoSH

4. Queries

Write SQL statements to answer the following queries. Write a comment header right before each SQL statement with the letter of the query answered by the statement.

- the total number of zones.
- the number of flowers per cool zone.
- common names of the plants that have delivery sizes less than 5.
- common names of the plants that require full sun (i.e., sun needs contains 'S').
- all delivery category names order alphabetically (without repetition).
- the exact output (note that it is order by Name):

Name	Cool Zone (low)	Cool Zone (high)	Delivery Category
Black Eyed Susan	-30	-20	pot
Lady Fern	-50	-40	plant
Lily-of-the-Valley	-50	-40	plant
Lombardy Poplar	-40	-30	tree
Nikko Blue Hydrangea	-20	-10	pot
Pink Caladiums	30	40	bulb
Purple Leaf Plum Hedge	-50	-40	hedge
Purple Liatris	-40	-30	bulb
Thorndale Ivy	-40	-30	pot
Variegated Weigela	-30	-20	shrub

g) plant names that have the same hot zone as “Pink Caladiums” (your solution MUST get the hot zone of “Pink Caladiums” in a variable).

h) the total number of plants, the minimum delivery size, the maximum delivery size, and the average size based on the plants that have delivery sizes (note that the average value should be rounded using two decimals).

Total	Min	Max	Average
6	1.5	36	14.33

i) the Latin name of the plant that has the word ‘Eyed’ in its name (you must use LIKE in this query to get full credit).

j) the exact output (ordered by Category and then by Name):

Category	Name
bulb	Pink Caladiums
bulb	Purple Liatris
hedge	Purple Leaf Plum Hedge
plant	Lady Fern
plant	Lily-of-the-Valley
pot	Black Eyed Susan
pot	Nikko Blue Hydrangea
pot	Thorndale Ivy
shrub	Variegated Weigela
tree	Lombardy Poplar

5. Rubric

This programming assignment is worth 100 points, distributed in the following way:

- +5 database, tables, and SQL file have the names suggested, comments were added on the SQL file, including the name of the student(s)
- +5 table Zones creation
- +5 table Zones population
- +5 table Deliveries creation
- +5 table Deliveries population
- +10 table FlowersInfo creation
- +5 table FlowersInfo population
- +25 queries a, b, d, e, and i, +5 each
- +35 +7 for the other queries

10 points will be deducted for each day late. I will not accept submissions that are five days (or more) late.