

## Wine program using MySQL:

For this lab, you will use mysql to store all the wine information into the mysql database instead of reading from a file.

If you are using the image OS file I provided at the beginning of semester, all the wine information already been stored in the MySQL database.

If you are NOT using the image OS file I provided, you must install mySQL and load the wine info into the database in your own environment. Please refer to mysql-createdb.pdf file for more information on how to create mysql database, wineInfo table and load winelist.txt into the wineInfo table if you are not using the image file I provided.

wineInfo.cpp and winedb.cpp (in wineDB-Lab.zip file): You can modify the sample program winedb.cpp and wineInfo for this lab. Program winedb.cpp has all the basic information on how to open the SQL database, and read the wine information from the wine table in mySQL database. You need to implement wine class object functions wine constructor and printWineInfo.

For this C++ wine program, you need read the wine info from the SQL database, and include the features below. Read the wine info from MySQL and then put them into vector:

**Vector<Wine> wineList;**

### **For Outputs: (MUST INCLUDE YOUR NAME AS PART OF THE PROGRAM OUTPUT)**

- Use a user menu to display user sections to until user select "Quit":
  1. Display all the wine has a score (valid input is between 0 and 100) between x and y. The output must be sorted by price. For example, if user enters range between 90 and 100, also display the total number of wine between the selected range and average price for the selected wines. **You need to check the validity of user's input.**

**Output should be similar as below:**

Wine Name	Vintage	Score	Price	Type
Stags Leap Chardonnay	2014	90	30	White
Grgich Chardonnay	2013	90	43	White
Stags Leap Artemis Cabernet	2013	92	65	Red
Alpha Omega Chardonnay	2012	92	69.99	White
Duckhorn Cabernet	2013	93	72	Red
Silver Oak Cabernet	2011	91	110	Red
Joseph Phelps Insignia	2013	97	240	Red
Opus One Bordeaux	2012	97	399.99	Red
Total number of wine found: 8				
Average price: \$128.75				

2. Display all the wine has a price between x and y, and after print out all the wine in the selected range, also print out the total number of wine between the selected range and average price for the selected wines. **Test case, if user enters range between 20 and 50, only wines with the**

price between 20 and 50 and sorted by price will be printed. You need to check the validity of user's input. Output should be similar as below:

Wine Name	Vintage	Score	Price	Type
-----	-----	-----	-----	----
Stags Leap Chardonnay	2014	90	30	White
Grgich Chardonnay	2013	90	43	White
Stags Leap Artemis Cabernet	2013	92	65	Red
Alpha Omega Chardonnay	2012	92	69.99	White
Duckhorn Cabernet	2013	93	72	Red
Total number of wine found: 5				
Average price: \$56.00				

3. Display all the wines that is sorted by score wine type and price. Highest score wine should be listed 1<sup>st</sup>. **Output should be similar to below:**

Wine Name	Vintage	Score	Price	Type
-----	-----	-----	-----	----
Opus One Bordeaux	2012	97	399.99	Red
Joseph Phelps Insignia	2013	97	240	Red
Duckhorn Cabernet	2013	93	72	Red
Stags Leap Artemis Cabernet	2013	92	65	Red
Alpha Omega Chardonnay	2012	92	69.99	White
Silver Oak Cabernet	2011	91	110	Red
Stags Leap Chardonnay	2014	90	30	White
Grgich Chardonnay	2013	90	43	White
Total number of wine found: 8				
Average price: \$128.75				

4. Quit – Exit program

- Program output must be neat and nicely formatted.

**Our lab assistants**, are available during the open lab hours. Don't wait until last minute to ask for help.

As part of submission is screen shots of the program output shows if the program is working or now.

**You need to submit two separate files:**

1. **The pdf file must contain**
  - Your name must be part of the program output (printMeFirst)
  - Source codes
  - All the screen shots of program output which include all the required outputs
  - Code must be clearly documented.
2. The source codes in zip file (all the files required to compile your program)

Points will be deducted if:

- if not all required screen shorts are submitted.
- program is not well documented
- didn't follow the lab requirements

For this lab, you need to get (read) all the wine information from mysql database. Use the sample dbconnect.cpp and windb.cpp files for this program.

To access MySQL on the image OS I provided: ( You don't need to do this as part of lab. However, I provided the example, so you can access the MySQL database directly if you want to use MySQL database in the future)

**mysql -u root -p**

Password: **password**

```
cs:wineDB-Lab$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.17-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use wine;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

Database schema:

```
mysql> show tables;
+-----+
| Tables_in_wine |
+-----+
| wineInfo       |
| wineryInfo     |
+-----+
2 rows in set (0.00 sec)

mysql> describe wineInfo;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name  | varchar(50) | YES | | NULL | |
| vintage | int(11) | YES | | NULL | |
| score | int(11) | YES | | NULL | |
| price | double | YES | | NULL | |
| wineryID | int(11) | YES | | NULL | |
| type  | varchar(20) | YES | | NULL | |
| location | varchar(50) | YES | | NULL | |
| UPC   | varchar(12) | NO | PRI | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

You need to use SQL select statement in your C++ program to retrieve data from the mysql database.

You can use the sample code in I provided

<https://drive.google.com/open?id=1bPJANTu-SlxQOXsBQcvlxf-xspXt9Z2>

Below is a sample of SQL select statement:

#### Get all the Red wines

*select \* from wineInfo where type = 'Red';*

```
mysql> select * from wineInfo where type = 'Red';
+-----+-----+-----+-----+-----+-----+-----+-----+
| name                | vintage | score | price | wineryID | type | location | UPC      |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Stags Leap Artemis Cabernet | 2013    | 92    | 65    | 1        | Red  | Stags    | 088593700200 |
| Opus One Bordeaux         | 2012    | 97    | 399.99 | 7        | Red  | Opus     | 5931942      |
| Duckhorn Cabernet         | 2013    | 93    | 72    | 4        | Red  | Duckhorn | 669576019344 |
| Silver Oak Cabernet       | 2011    | 91    | 110   | 2        | Red  | Silver Oak | 892026001006 |
| Joseph Phelps Insignia    | 2013    | 97    | 240   | 3        | Red  | Joseph   | 99999        |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

#### Get all the wines that are not Red

*Select \* from wineInfo where type != 'Red';*

#### Get all the wines that the vintage is between 2009 and 2010

*Select \* from wineInfo where vintage >= 2009 and vintage <= 2010;*

#### Get all the wines that the rating is between 90 and 92

*Select \* from wineInfo where score >= 90 and score <=92;*

#### Get all the wines that the price is above 100 and sorted by price with highest price listed 1st

*Select \* from wineInfo where price > 100 order by price ASC;*

#### Get all the wines and sorted by wine type, price with lowest price 1st

*Select \* from wineInfo order by type, price DESC;*

```
mysql> select * from wineInfo order by type, price DESC;
+-----+-----+-----+-----+-----+-----+-----+-----+
| name                | vintage | score | price | wineryID | type | location | UPC      |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Opus One Bordeaux         | 2012    | 97    | 399.99 | 7        | Red  | Opus     | 5931942      |
| Joseph Phelps Insignia    | 2013    | 97    | 240    | 3        | Red  | Joseph   | 99999        |
| Silver Oak Cabernet       | 2011    | 91    | 110    | 2        | Red  | Silver Oak | 892026001006 |
| Duckhorn Cabernet         | 2013    | 93    | 72     | 4        | Red  | Duckhorn | 669576019344 |
| Stags Leap Artemis Cabernet | 2013    | 92    | 65     | 1        | Red  | Stags    | 088593700200 |
| Alpha Omega Chardonnay    | 2012    | 92    | 69.99  | 5        | White | Alpha Omega | 99988        |
| Grgich Chardonnay        | 2013    | 90    | 43     | 6        | White | Grgich   | 1271957      |
| Stags Leap Chardonnay     | 2014    | 90    | 30     | 1        | White | Stags    | 089819043019 |
+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```