

LAB: CS 116 Lab

NAME: Haichuan Wei

DATE:12/5/2021

Program Description:

This program practices the use of MYSQL databases. This lab only practices the basics of using the database. Loading, creating, and displaying. It uses MYSQL server database to store the data. The user can choose between many different option, sorting it, adding to the database, change the data base, and quitting.

Source Code:

Part 1: Used only provided Files

Part 2: Used only provided files

Part 3:

winedb.cpp

```
1 #include "printMeFirst.h"
2 #include <usr/include/mysql/mysql.h>
3 #include "winedata.h"
4 #include "dbconnect.h"
5 using namespace std;
6
7 int main(int argc, char *argv[])
8 {
9     printMeFirst("Haichuan Wei", "CS116 Lab 10");
10    //connect to the database
11    MYSQL *conn; // the connection
12    MYSQL_RES *res; // the results
13    MYSQL_ROW fields; // the results row (line by line)
14    struct connection_details mysqlDb;
15    mysqlDb.server = (char *)"localhost"; // where the mysql database is
16    mysqlDb.user = (char *)"cs116"; // the root user of mysql
17    mysqlDb.password = (char *)"OhloneCS116"; // the password of the user in mysql
18    mysqlDb.database = (char *)"mysql"; // the database to pick
19    // connect to the mysql database
20    conn = mysql_connection_setup(mysqlDb);
21    res = mysql_perform_query(conn, (char *)"use cs116");
22
23    // perform the query
24    cout << "Welcome to the wine database!" << endl;
25    cout << "Please select an option:" << endl;
26    cout << "1. Display wine Between two scores" << endl;
27    cout << "2. Display wine between two prices" << endl;
28    cout << "3. Display top 10 wine" << endl;
29    cout << "4. Insert New Wine" << endl;
30    cout << "5. Update an existing Wine data" << endl;
31    cout << "6. Quit" << endl;
32    int selection;
33    cin >> selection;
34    while (selection != 6)
35    {
36        if (selection == 1)
37        {
38            string query = displayscore();
39            res = mysql_perform_query(conn, (char *)query.c_str());
40            printwine(res);
41            cout << query << endl;
42        }
43        else if (selection == 2)
44        {
45            string query = displayprice();
46            res = mysql_perform_query(conn, (char *)query.c_str());
47            printwine(res);
48            cout << query << endl;
49        }
50    }
```

```
Intermediate C++ > Lab 10 > C- winedb.cpp > main(int, char* []  
48     cout << query << endl;  
49 }  
50 else if (selection == 3)  
51 {  
52     string query = topten();  
53     res = mysql_perform_query(conn, (char *)query.c_str());  
54     printwine(res);  
55     cout << query << endl;  
56 }  
57 else if (selection == 4)  
58 {  
59     string query = insertwine();  
60     mysql_perform_query(conn, (char *)query.c_str());  
61     res = mysql_perform_query(conn, (char *)" SELECT * FROM wineInfo ORDER BY upc DESC LIMIT 1; ");  
62     printwine(res);  
63     cout << query << endl;  
64 }  
65 else if (selection == 5)  
66 {  
67     string query = updatewine();  
68     mysql_perform_query(conn, (char *)query.c_str());  
69     res = mysql_perform_query(conn, (char *)" SELECT * FROM wineInfo WHERE price = '399' ");  
70     printwine(res);  
71     cout << query << endl;  
72 }  
73 else  
74 {  
75     cout << "Invalid selection" << endl;  
76 }  
77 while (cin.fail() || selection < 1 || selection > 6)  
78 {  
79     cout << "Invalid input, please enter a number between 1 and 6" << endl;  
80     cin.clear();  
81     cin.ignore(256, '\n');  
82     cin >> selection;  
83 }  
84 cout << "Please select an option:" << endl;  
85 cout << "1. Display wine Between two scores" << endl;  
86 cout << "2. Display wine between two prices" << endl;  
87 cout << "3. Display top 10 wine" << endl;  
88 cout << "4. Insert New Wine" << endl;  
89 cout << "5. Update an existing Wine data" << endl;  
90 cout << "6. Quit" << endl;  
91 cin >> selection;  
92 }  
93 /* Clean up the database link */  
94 mysql_close(conn);  
95 return 0;  
96 }
```

winedata.cpp

```
Intermediate C++ > Lab 10 > C- winedata.cpp > updatewine()  
1  
2 #include "winedata.h"  
3 using namespace std;  
4 //print header  
5 void printHeader()  
6 {  
7     cout << " " << setw(25) << "Wine Name"  
8     << setw(23) << "Vintage"  
9     << setw(9) << "Rating"  
10    << setw(7) << "Price"  
11    << setw(9) << "Type"  
12    << endl;  
13    cout << " " << setw(25) << "-----"  
14    << setw(23) << "-----"  
15    << setw(9) << "-----"  
16    << setw(7) << "-----"  
17    << setw(9) << "-----"  
18    << endl;  
19 }  
20  
21 void printwine(MYSQL_RES *res)  
22 {  
23     MYSQL_ROW row;  
24     int count = 0;  
25     double price = 0;  
26     double whiteprice = 0;  
27     double redprice = 0;  
28     int redcount = 0;  
29     int whitecount = 0;  
30  
31     //print the result of the sql query  
32     printHeader();  
33     while ((row = mysql_fetch_row(res)) != NULL)  
34     {  
35         string type = row[4];  
36         count++;  
37         price += atof(row[3]);  
38         /* print out each row of the data extracted from  
39          * MySQL database  
40          * Make sure the output is line up with the header  
41          * Hint: use left and setw  
42          */  
43  
44         cout << " " << setw(42) << left << row[0] << " " // column (field) #1 - Wine Name  
45         << setw(6) << row[1] << " " // field #2 - Vintage  
46         << setw(7) << row[2] << " " // field #3 - Rating/Score  
47         << setw(8) << row[3] << " " // field #4 - Price  
48         << setw(8) << row[4] << " " // field #5 - Wine type  
49         << endl;  
50     }  
51 }
```

```

Intermediate C++ > Lab 10 > C- winedata.cpp > updatewine()
46     << setw(5) << row[4] << // field #5 - wine type
47     << endl; //
50     if (type == "white" || type == "white")
51     {
52         whitecount++;
53         whiteprice += atof(row[3]);
54     }
55     else if (type == "Red" || type == "red")
56     {
57         redcount++;
58         redprice += atof(row[3]);
59     }
60 }
61 cout << "\n\n";
62 << endl;
63 cout << "Number of wines: " << count << " Average price: $"
64 << fixed << setprecision(2) << price / count << endl;
65 cout << "Number of red wines: " << redcount << " Average price: $"
66 << fixed << setprecision(2) << redprice / redcount << endl;
67 cout << "Number of white wines: " << whitecount << " Average price: $"
68 << fixed << setprecision(2) << whiteprice / whitecount << endl;
69 cout << "\n\n";
70 << endl;
71 };
72
73 string displayprice()
74 {
75     int x, y;
76     cout << "Please enter the lower bound of price (0):" << endl;
77     cin >> x;
78     if (cin.fail() || x < 0)
79     {
80         cin.clear();
81         cin.ignore(100, '\n');
82         cout << "Invalid input, please enter again: ";
83         cin >> x;
84     }
85
86     cout << "Please enter the upper bound of price(100):" << endl;
87     cin >> y;
88     if (cin.fail() || y < 0 || x > y)
89     {
90         cin.clear();
91         cin.ignore(100, '\n');
92         cout << "Invalid input, please enter again: ";
93         cin >> y;
94     }
95     string query = "SELECT * FROM wineInfo WHERE price BETWEEN " + to_string(x) + " AND " + to_string(y) + " order by price DESC , score DESC ";
96     return query;
97 }

```

```

Intermediate C++ > Lab 10 > C- winedata.cpp > updatewine()
146     string query = "INSERT INTO wineInfo (name, vintage, score, price, type, upc) VALUES ('" + name + "', " + to_string(vintage) + ", " + to_string(score) + ", " + to_string(pr
147     return query;
148 };
149 string updatewine()
150 {
151     string name, type;
152     int vintage, score, price, upc;
153     cout << "which data do you want use to update other data?" << endl;
154     cout << "1. name" << endl;
155     cout << "2. upc" << endl;
156     int onetwo;
157     cin >> onetwo;
158     if (cin.fail() || onetwo < 1 || onetwo > 2)
159     {
160         cin.clear();
161         cin.ignore(100, '\n');
162         cout << "Invalid input, please enter again: ";
163         cin >> onetwo;
164     }
165     switch (onetwo)
166     {
167     case 1:
168     {
169         cout << "Please enter the name of the wine: " << endl;
170         cin >> name;
171         break;
172     }
173     case 2:
174     {
175         cout << "Please enter the upc of the wine: " << endl;
176         cin >> upc;
177         break;
178     }
179     }
180     cout << "\n\n 1. name" << endl;
181     cout << "2. vintage" << endl;
182     cout << "3. score" << endl;
183     cout << "4. price" << endl;
184     cout << "5. type" << endl;
185     cout << "6. upc" << endl;
186     int choice;
187     cin >> choice;
188     if (cin.fail() || choice < 1 || choice > 6)
189     {
190         cin.clear();
191         cin.ignore(100, '\n');
192         cout << "Invalid input, please enter again: ";
193         cin >> choice;
194     }

```

Intermediate C++ > Lab 10 > C++ wineData.cpp > updatewine()

```
196 {
197     case 1:
198         cout << "Please enter the new name of the wine: " << endl;
199         cin >> name;
200         break;
201     case 2:
202         cout << "Please enter the new vintage of the wine: " << endl;
203         cin >> vintage;
204         break;
205     case 3:
206         cout << "Please enter the new score of the wine: " << endl;
207         cin >> score;
208         break;
209     case 4:
210         cout << "Please enter the new price of the wine: " << endl;
211         cin >> price;
212         break;
213     case 5:
214         cout << "Please enter the new type of the wine: " << endl;
215         cin >> type;
216         break;
217     case 6:
218         cout << "Please enter the new upc of the wine: " << endl;
219         cin >> upc;
220         break;
221 }
222 string query = "UPDATE wineInfo SET ";
223 switch (choice)
224 {
225     case 1:
226         query += "name = '" + name + "' ";
227         break;
228     case 2:
229         query += "vintage = " + to_string(vintage) + " ";
230         break;
231     case 3:
232         query += "score = " + to_string(score) + " ";
233         break;
234     case 4:
235         query += "price = " + to_string(price) + " ";
236         break;
237     case 5:
238         query += "type = '" + type + "' ";
239         break;
240     case 6:
241         query += "upc = " + to_string(upc) + " ";
242         break;
243 }
244 query += "WHERE ";
245
246 query += "vintage = " + to_string(vintage) + " ";
247 break;
248 case 3:
249     query += "score = " + to_string(score) + " ";
250     break;
251 case 4:
252     query += "price = " + to_string(price) + " ";
253     break;
254 case 5:
255     query += "type = '" + type + "' ";
256     break;
257 case 6:
258     query += "upc = " + to_string(upc) + " ";
259     break;
260 }
261 query += "WHERE ";
262 switch (onetwo)
263 {
264     case 1:
265         query += "name = '" + name + "' ";
266         break;
267     case 2:
268         query += "upc = " + to_string(upc) + " ";
269         break;
270 }
271 return query;
272 };
```

Winedata.h

```
Intermediate C++ > Lab 10 > | winedata.h > ...
1  /*
2  Purpose- This is the header file. It calculates the wine data, displays it, and it can be used to calculate the average. All data is fed from the my sql server.
3  @author Haichuan Wei
4  @version 1.0 11/14/21
5
6  @function printHeader() - This function prints the table header
7  @function printwine() - This function prints every wine based on the input
8  @function displayscore() - This function displays the score of the wine based on the minimum and maximum score
9  @function displayprice() - This function displays the price of the wine based on the minimum and maximum price
10 @function topten() - This function displays the top ten wines sorted by price
11 @function insertwine() - This function inserts a wine into the database
12 @function updatewine() - This function updates a wine in the database
13
14 */
15
16 #ifndef _winedata_h
17 #define _winedata_h
18 #include <usr/include/mysql/mysql.h>
19 #include <iostream>
20 #include <string>
21 #include <vector>
22 #include "dbconnect.h"
23
24 using namespace std;
25
26 void printHeader();
27 void printwine(MYSQL_RES *res);
28 string displayscore();
29 string displayprice();
30 string topten();
31 string insertwine();
32 string updatewine();
33
34 #endif // !_winedata_h
35
```

No need to document Dbconnect.h/ Dbconnect.cpp.

Program Output:

Part 1: Output matches Example

```
mysql> use cs116
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> LOAD DATA LOCAL INFILE 'winelist.txt' INTO TABLE wineInfo FIELDS TERMINATED BY ';';
Query OK, 100 rows affected, 100 warnings (0.02 sec)
Records: 100 Deleted: 0 Skipped: 0 Warnings: 100

mysql> LOAD DATA LOCAL INFILE 'winelist.txt' INTO TABLE wineInfo FIELDS
-> TERMINATED BY ';' ;
Query OK, 0 rows affected, 200 warnings (0.00 sec)
Records: 100 Deleted: 0 Skipped: 100 Warnings: 200

mysql> SELECT * FROM wineInfo where price > 100;
+-----+-----+-----+-----+-----+-----+-----+
| name                                     | vintage | score | price | type  | location | UPC   |
+-----+-----+-----+-----+-----+-----+-----+
| Chateau Leoville Las Cases St.-Julien   | 2011    | 95    | 165   | Red   | location | 10    |
| Fonseca Vintage Port                    | 2011    | 98    | 116   | Red   | location | 13    |
| Fontodi Colli della                     | 2011    | 95    | 120   | Red   | location | 14    |
| Marcassin Pinot Noir Sonoma Marcassin   | 2009    | 97    | 125   | White | location | 36    |
| Luce della Vite Toscana Luce            | 2011    | 95    | 105   | White | location | 47    |
| Giuseppe Rinaldi Barolo Brunate         | 2010    | 97    | 129   | White | location | 51    |
| TwentyFour Cabernet Sauvignon Napa Valley | 2010    | 93    | 112   | White | location | 58    |
| Ornellaia Bolgheri Superiore            | 2011    | 96    | 240   | White | location | 65    |
| Clos des Papes Chateauneuf-du-Pape     | 2012    | 97    | 135   | Red   | location | 7     |
| E. Guigal Cote-Rotie Chateau d'Ampuis  | 2010    | 97    | 206   | White | location | 77    |
| Concha y Toro Cabernet Sauvignon        | 2010    | 95    | 125   | Red   | location | 9     |
+-----+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> |
```

Part 2: Output matches Example

```
mysql> SELECT name, vintage, score, price, type FROM wineInfo WHERE vintage >= 2009 and
-> vintage <= 2010;
```

name	vintage	score	price	type
Castello di Volpaia Chianti	2010	93	29	Red
St.-Cosme Chateauneuf-du-Pape	2010	96	59	Red
Massolino Barolo	2009	95	60	Red
Bodegas Fournier Malbec	2010	94	54	Red
Hidden Ridge Cabernet Sauvignon	2009	93	45	White
Marcassin Pinot Noir Sonoma Marcassin	2009	97	125	White
Giuseppe Cortese Barbaresco Rabaja	2010	94	55	White
Oddero Barolo	2010	93	49	White
Lapostolle Clos Apalta	2010	94	89	White
Abadia Retuerta Vino Duero	2010	92	35	White
Giuseppe Rinaldi Barolo Brunate	2010	97	129	White
TwentyFour Cabernet Sauvignon Napa Valley	2010	93	112	White
Castello di Ama Chianti	2010	95	52	Red
Bodegas Montecillo Rioja Vina	2010	90	10	White
Godelia Mencia Bierzo	2010	92	21	White
Feudo di Santa Croce Primitivo di	2010	91	22	White
E. Guigal Cote-Rotie Chateau d'Ampuis	2010	97	206	White
Marques de Grignon Cabernet Sauvignon	2010	93	40	White
Nino Negri Valtellina Superiore Quadrio	2010	90	20	White
Domaine de Triennes Vin de Pays du	2010	90	20	White
Masciarelli Montepulciano d'Abruzzo	2010	91	25	White
Pali Wine Co. Pinot Noir Santa Barbara	2010	90	23	White
Concha y Toro Cabernet Sauvignon	2010	95	125	Red
Vecchia Cantina di Montepulciano Vino	2010	90	15	White

24 rows in set (0.00 sec)

```
mysql> SELECT name, vintage, score, price, type FROM wineInfo WHERE price > 100 order by price
-> DESC;
```

name	vintage	score	price	type
Ornellaia Bolgheri Superiore	2011	96	240	White
E. Guigal Cote-Rotie Chateau d'Ampuis	2010	97	206	White
Chateau Leoville Las Cases St.-Julien	2011	95	165	Red
Clos des Papes Chateauneuf-du-Pape	2012	97	135	Red
Giuseppe Rinaldi Barolo Brunate	2010	97	129	White
Marcassin Pinot Noir Sonoma Marcassin	2009	97	125	White
Concha y Toro Cabernet Sauvignon	2010	95	125	Red
Fontodi Colli della	2011	95	120	Red
Fonseca Vintage Port	2011	98	116	Red
TwentyFour Cabernet Sauvignon Napa Valley	2010	93	112	White
Luce della Vite Toscana Luce	2011	95	105	White

11 rows in set (0.00 sec)

```
mysql> exit
```

Part 3:

Program 1:

Test Case 1: Matched Example

```
Welcome to the wine database!
Please select an option:
1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit
1
Please enter the lower bound of price:
nintynine
Invalid input, please enter again: 99
Please enter the upper bound of price:
onehundred
Invalid input, please enter again: 100
SELECT * FROM wineInfo WHERE price BETWEEN 99 AND 100 order by price DESC , score DESC
```

Test case 2: Matched Example

```
./winedb "select * from wineInfo where price > 100"
Program written by: Haichuan Wei
Course Info: CS116 Lab 10
Date: Sun Dec 5 23:01:50 2021

Welcome to the wine database!
Please select an option:
1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit
1
Please enter the lower bound of score(0>):
120
Invalid input, please enter again: 50
Please enter the upper bound of score(100<):
200
Invalid input, please enter again: 90
```

Wine Name	Vintage	Rating	Price	Type
Round Pond Estate Sauvignon Blanc	2013	90	24	White

Test case 3: Matched Example

Program written by: Haichuan Wei
Course Info: CS116 Lab 10
Date: Sun Dec 5 23:04:47 2021

Welcome to the wine database!

Please select an option:

1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit

1

Please enter the lower bound of score(0>):

96

Please enter the upper bound of score(100<):

97

Wine Name	Vintage	Rating	Price	Type
-----	-----	-----	-----	-----
St.-Cosme Chateauneuf-du-Pape	2010	96	59	Red
Leeuwin Chardonnay River Art Series	2011	96	89	Red
Ornellaia Bolgheri Superiore	2011	96	240	White
Chateau Guiraud Sauternes	2011	97	50	Red
Prats & Symington Douro Chryseia	2011	97	55	Red
Marcassin Pinot Noir Sonoma Marcassin	2009	97	125	White
Quinta do Vale Meao Douro	2011	97	76	Red
Giuseppe Rinaldi Barolo Brunate	2010	97	129	White
Clos des Papes Chateauneuf-du-Pape	2012	97	135	Red
E. Guigal Cote-Rotie Chateau d'Ampuis	2010	97	206	White

Number of wines: 10 Average price: \$116.40

Number of red wines: 6 Average price: \$77.33

Number of white wines: 4 Average price: \$175.00

SELECT * FROM wineInfo WHERE score BETWEEN 96 AND 97 order by score ASC

Please select an option:

1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine

Program 2: Matched Example

```
Program written by: Haichuan Wei
Course Info: CS116 Lab 10
Date: Sun Dec 5 23:08:10 2021

Welcome to the wine database!
Please select an option:
1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit
2
Please enter the lower bound of price (0>):
-3
Invalid input, please enter again: 100
Please enter the upper bound of price(100<):
two
Invalid input, please enter again: 200
```

Wine Name	Vintage	Rating	Price	Type
-----	-----	-----	-----	-----
Chateau Leoville Las Cases St.-Julien	2011	95	165	Red
Clos des Papes Chateauneuf-du-Pape	2012	97	135	Red
Giuseppe Rinaldi Barolo Brunate	2010	97	129	White
Marcassin Pinot Noir Sonoma Marcassin	2009	97	125	White
Concha y Toro Cabernet Sauvignon	2010	95	125	Red
test	123	123	123	123
Fontodi Colli della	2011	95	120	Red
Fonseca Vintage Port	2011	98	116	Red
TwentyFour Cabernet Sauvignon Napa Valley	2010	93	112	White
Luce della Vite Toscana Luce	2011	95	105	White

```
Number of wines: 10    Average price: $125.50
Number of red wines: 5    Average price: $132.20
Number of white wines: 4    Average price: $117.75
```

```
SELECT * FROM wineInfo WHERE price BETWEEN 100 AND 200 order by price DESC , score DESC
Please select an option:
1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
```

Program 3: Matched Example (I added some wines to test program 4)

Please select an option:

1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit

3

Wine Name	Vintage	Rating	Price	Type	
-----	-----	-----	-----	-----	
dfg		234	234	543	12
OpusOne		2016	98	369	Red
OpusOne		2016	98	369	Red
Test		3954	324	324	Red
Ornellaia Bolgheri Superiore		2011	96	240	White
E. Guigal Cote-Rotie Chateau d'Ampuis		2010	97	206	White
Chateau Leoville Las Cases St.-Julien		2011	95	165	Red
Clos des Papes Chateauneuf-du-Pape		2012	97	135	Red
Giuseppe Rinaldi Barolo Brunate		2010	97	129	White
Marcassin Pinot Noir Sonoma Marcassin		2009	97	125	White

Number of wines: 10 Average price: \$260.50

Number of red wines: 5 Average price: \$272.40

Number of white wines: 4 Average price: \$175.00

SELECT * FROM wineInfo order by price DESC limit 10

Please select an option:

1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit

|

Program 4: Matched Example

```
Program written by: Haichuan Wei
Course Info: CS116 Lab 10
Date: Sun Dec 5 23:44:42 2021

Welcome to the wine database!
Please select an option:
1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit
4
Please enter the name of the wine:
OpusOne
Please enter the vintage of the wine:
2016
Please enter the score of the wine:
98
Please enter the price of the wine:
369
Please enter the type of the wine:
Red
Please enter the upc of the wine:
9999

      Wine Name      Vintage  Rating  Price   Type
      -----      -
OpusOne              2016     98     369     Red

-----

Number of wines: 1    Average price: $369.00
Number of red wines: 1    Average price: $369.00
Number of white wines: 0    Average price: $-nan

-----

INSERT INTO wineInfo (name, vintage, score, price, type, upc) VALUES ('OpusOne', 2016, 98, 369, 'Red', 9999)
Please select an option:
1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
```

Program 5: Matched Example

```
Program written by: Haichuan Wei
Course Info: CS116 Lab 10
Date: Sun Dec 5 23:45:53 2021

Welcome to the wine database!
Please select an option:
1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit
5
which data do you want use to update other data?
1. name
2. upc
2
Please enter the upc of the wine:
9999

1. name
2. vintage
3. score
4. price
5. type
6. upc
4
Please enter the new price of the wine:
399

      Wine Name      Vintage  Rating  Price  Type
      -----
OpusOne             2016      98      399    Red

Number of wines: 1    Average price: $399.00
Number of red wines: 1    Average price: $399.00
Number of white wines: 0    Average price: $-nan

UPDATE wineInfo SET price = 399 WHERE upc = 9999
Please select an option:
1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit
|
```

Quit Case:

```
UPDATE wineInfo SET price = 399 WHERE upc = 9999
Please select an option:
1. Display wine Between two scores
2. Display wine between two prices
3. Display top 10 wine
4. Insert New Wine
5. Update an existing Wine data
6. Quit
6
arthur@DESKTOP-UP5LF24:~/Cpp_Projects-1/Intermediate C++/Lab 10$ |
```