Group 2

Bacchus Winery Case Study

Revised ERD



Screenshot(s) of the data displayed from tables

**Created a new database called bacchus\_winery in MySQL Command Line Client**

**INITIAL PYTHON SCRIPT**

The following code is from the example from <https://dev.mysql.com/doc/connector-python/en/connector-python-example-ddl.html>

from \_\_future\_\_ import print\_function

import mysql.connector

from mysql.connector import errorcode

config = {

'user': 'root',

'password': 'P!aybill!985',

'host': 'localhost',

'database': 'bacchus\_winery',

'raise\_on\_warnings': True

}

try:

mydb = mysql.connector.connect(\*\*config)

print("\n Database user {} connected to MySQL on host {} with database {}".format(

config["user"], config["host"], config["database"]))

input("\n\n Press any key to continue...\n")

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_ACCESS\_DENIED\_ERROR:

print(" The supplied username or password are invalid")

elif err.errno == errorcode.ER\_BAD\_DB\_ERROR:

print(" The specified database does not exist")

else:

print(err)

DB\_NAME = 'bacchus\_winery'

myCursor = mydb.cursor()

# Function to erase tables if they exist to start with a clean database

def drop\_tables(myCursor):

tables = "DROP TABLE wine"

myCursor.execute(tables)

mydb.commit()

drop\_tables(myCursor)

TABLES = {}

TABLES['wine'] = (

"CREATE TABLE wine"

" (wine\_id int NOT NULL AUTO\_INCREMENT KEY,"

" wine\_name varchar(25) NOT NULL,"

" units int NOT NULL,"

" batch\_month varchar(25) NOT NULL )")

'''TABLES['orders'] = (

"CREATE TABLE orders"

" (order\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" units int NOT NULL,"

" wine\_nam varchar(25) NOT NULL,"

" CONSTRAINT fk\_wine FOREIGN KEY (units) REFERENCES wine(units))")

TABLES['distributors'] = (

"CREATE TABLE distributors"

" (distributor\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" distributor\_name varchar(25) NOT NULL,"

" order\_id int NOT NULL,"

" CONSTRAINT fk\_orders FOREIGN KEY (order\_id) REFERENCES orders (order\_id))")

TABLES['suppliers'] = (

"CREATE TABLE suppliers"

" (supplier\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" supplier\_name varchar(25) NOT NULL,"

" expected\_delivery\_date DATE NOT NULL,"

" actual\_delivery\_date DATE NOT NULL,"

" supply\_name varchar(25) NOT NULL)")

TABLES['deliveries'] = (

"CREATE TABLE deliveries"

" (delivery\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" supplier\_id int NOT NULL,"

" inventory int NOT NULL,"

" CONSTRAINT fk\_suppliers FOREIGN KEY (supplier\_id) REFERENCES suppliers(supplier\_id))")

TABLES['supplies'] = (

"CREATE TABLE supplies"

" (supply\_id int NOT NULL PRIMARY KEY,"

" supply\_name varchar(25) NOT NULL,"

" inventory int NOT NULL,"

" CONSTRAINT fk\_suppliers FOREIGN KEY (supply\_name) REFERENCES suppliers(supplier\_name),"

" CONSTRAINT fk\_deliveries FOREIGN KEY (inventory) REFERENCES deliveries(inventory))")

TABLES['employees'] = (

"CREATE TABLE employees"

" (employee\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" first\_name varchar(25) NOT NULL,"

" last\_name varchar(25) NOT NULL,"

" job\_title varchar(25) NOT NULL)")

TABLES['time\_sheet'] = (

"CREATE TABLE time\_sheet"

" (week\_id int NOT NULL PRIMARY KEY,"

" hours\_worked\_weekly int NOT NULL,"

" ot\_worked\_weekly int NOT NULL,"

" employee\_id int NOT NULL,"

" CONSTRAINT fk\_employees FOREIGN KEY (employee\_id) REFERENCES employees (employee\_id))")

'''

for table\_name in TABLES:

table\_description = TABLES[table\_name]

try:

print("Creating table {}: ".format(table\_name), end='')

myCursor.execute(table\_description)

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_TABLE\_EXISTS\_ERROR:

print("already exists.")

else:

print(err.msg)

else:

print("OK\n")

# Inset values into wine table

sql1 = "INSERT INTO wine (wine\_name, units, batch\_month) VALUE ('Merlot', 50, 'January')"

sql2 = "INSERT INTO wine (wine\_name, units, batch\_month) VALUE ('Cabernet', 40, 'February')"

sql3 = "INSERT INTO wine (wine\_name, units, batch\_month) VALUE ('Chablis', 60, 'April')"

sql4 = "INSERT INTO wine (wine\_name, units, batch\_month) VALUE ('Chardonnay', 50, 'March')"

myCursor.execute(sql1)

myCursor.execute(sql2)

myCursor.execute(sql3)

myCursor.execute(sql4)

mydb.commit()

#Print out values

print("-Displaying Wine Records-")

query1 = "SELECT wine\_name, units, batch\_month FROM wine"

myCursor.execute(query1)

wines = myCursor.fetchall()

for wine in wines:

print("Wine Name: {}\nNumber of units: {}\nBatch Month: {}\n".format(wine[0], wine[1],

wine[2]))

**REVISED PYTHON SCRIPT**

**(WORK IN PROGRESS)**

from \_\_future\_\_ import print\_function

import mysql.connector

from mysql.connector import errorcode

config = {

'user': 'root',

'password': 'Studman081!',

'host': 'localhost',

'database': 'bacchus\_winery',

'raise\_on\_warnings': True

}

try:

mydb = mysql.connector.connect(\*\*config)

print("\n Database user {} connected to MySQL on host {} with database {}".format(

config["user"], config["host"], config["database"]))

input("\n\n Press any key to continue...\n")

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_ACCESS\_DENIED\_ERROR:

print(" The supplied username or password are invalid")

elif err.errno == errorcode.ER\_BAD\_DB\_ERROR:

print(" The specified database does not exist")

else:

print(err)

DB\_NAME = 'bacchus\_winery'

myCursor = mydb.cursor()

# Function to erase tables if they exist to start with a clean database

'''def drop\_tables(myCursor):

tables1 = "DROP TABLE wine"

myCursor.execute(tables1)

mydb.commit()

drop\_tables(myCursor)'''

TABLES = {}

TABLES['wine'] = (

"CREATE TABLE wine"

" (wine\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" wine\_name varchar(25) NOT NULL,"

" units int NOT NULL,"

" batch\_month varchar(25) NOT NULL)")

TABLES['orders'] = (

"CREATE TABLE orders"

" (order\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" units int NOT NULL,"

" wine\_name varchar(25) NOT NULL)")

TABLES['distributors'] = (

"CREATE TABLE distributors"

" (distributor\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" distributor\_name varchar(25) NOT NULL,"

" order\_id int NOT NULL)")

TABLES['suppliers'] = (

"CREATE TABLE suppliers"

" (supplier\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" supplier\_name varchar(25) NOT NULL,"

" expected\_delivery\_date DATE NOT NULL,"

" actual\_delivery\_date DATE NOT NULL,"

" supply\_name varchar(25) NOT NULL)")

TABLES['deliveries'] = (

"CREATE TABLE deliveries"

" (delivery\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" supplier\_id int NOT NULL,"

" inventory int NOT NULL)")

TABLES['supplies'] = (

"CREATE TABLE supplies"

" (supply\_id int NOT NULL PRIMARY KEY,"

" supply\_name varchar(25) NOT NULL,"

" inventory int NOT NULL)")

TABLES['employees'] = (

"CREATE TABLE employees"

" (employee\_id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,"

" first\_name varchar(25) NOT NULL,"

" last\_name varchar(25) NOT NULL,"

" job\_title varchar(25) NOT NULL)")

TABLES['time\_sheet'] = (

"CREATE TABLE time\_sheet"

" (week\_id int NOT NULL PRIMARY KEY,"

" hours\_worked\_weekly int NOT NULL,"

" ot\_worked\_weekly int NOT NULL,"

" employee\_id int NOT NULL)")

for table\_name in TABLES:

table\_description = TABLES[table\_name]

try:

print("Creating table {}: ".format(table\_name), end='')

myCursor.execute(table\_description)

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_TABLE\_EXISTS\_ERROR:

print("already exists.")

else:

print(err.msg)

else:

print("OK\n")

# Insert values into wine table

sql1 = "INSERT INTO wine (wine\_name, units, batch\_month) VALUE ('Merlot', 50, 'January')"

sql2 = "INSERT INTO wine (wine\_name, units, batch\_month) VALUE ('Cabernet', 40, 'February')"

sql3 = "INSERT INTO wine (wine\_name, units, batch\_month) VALUE ('Chablis', 60, 'April')"

sql4 = "INSERT INTO wine (wine\_name, units, batch\_month) VALUE ('Chardonnay', 50, 'March')"

myCursor.execute(sql1)

myCursor.execute(sql2)

myCursor.execute(sql3)

myCursor.execute(sql4)

mydb.commit()

# Print out values

print("-Displaying Wine Records-")

query1 = "SELECT wine\_name, units, batch\_month FROM wine"

myCursor.execute(query1)

wines = myCursor.fetchall()

for wine in wines:

print("Wine Name: {}\nNumber of units: {}\nBatch Month: {}\n".format(wine[0], wine[1],

wine[2]))

# Insert values into Orders table

orders1 = "INSERT INTO orders (units, wine\_name) VALUE (10, 'Merlot')"

orders2 = "INSERT INTO orders (units, wine\_name) VALUE (15, 'Cabernet')"

orders3 = "INSERT INTO orders (units, wine\_name) VALUE (12, 'Chablis')"

orders4 = "INSERT INTO orders (units, wine\_name) VALUE (10, 'Chardonnay')"

myCursor.execute(orders1)

myCursor.execute(orders2)

myCursor.execute(orders3)

myCursor.execute(orders4)

mydb.commit()

print("-Displaying Orders-")

query2 = "SELECT units, wine\_name FROM orders"

myCursor.execute(query2)

orders = myCursor.fetchall()

for order in orders:

print("Number of units: {}\nWine Name: {}\n".format(order[0], order[1]))

# Insert values into Distributors table

dis1 = "INSERT INTO distributors (distributor\_name, order\_id) VALUE ('Wines to GO', 2)"

dis2 = "INSERT INTO distributors (distributor\_name, order\_id) VALUE ('Partners in Wine', 3)"

dis3 = "INSERT INTO distributors (distributor\_name, order\_id) VALUE ('Sip Happens Co.', 1)"

myCursor.execute(dis1)

myCursor.execute(dis2)

myCursor.execute(dis3)

mydb.commit()

print("-Displaying Distributors-")

query3 = "SELECT distributor\_name, order\_id FROM distributors"

myCursor.execute(query3)

distributors = myCursor.fetchall()

for distributor in distributors:

print("Distributor Name: {}\nOrder ID: {}\n".format(distributor[0], distributor[1]))

# Insert values into Suppliers table

sup1 = "INSERT INTO suppliers (supplier\_name, expected\_delivery\_date, actual\_delivery\_date, supply\_name)" \

"VALUE ('The New Corker', 20220110, 20220115, 'Corks')"

sup2 = "INSERT INTO suppliers (supplier\_name, expected\_delivery\_date, actual\_delivery\_date, supply\_name)" \

"VALUE ('Boxes n Stuff', 20220215, 20220215, 'Boxes')"

sup3 = "INSERT INTO suppliers (supplier\_name, expected\_delivery\_date, actual\_delivery\_date, supply\_name)" \

"VALUE ('Wine Supply', 20220320, 20220415, 'Tubing')"

myCursor.execute(sup1)

myCursor.execute(sup2)

myCursor.execute(sup3)

mydb.commit()

# Print out values

print("-Displaying Suppliers Records-")

query4 = "SELECT supplier\_name, expected\_delivery\_date, actual\_delivery\_date, supply\_name FROM suppliers"

myCursor.execute(query4)

suppliers = myCursor.fetchall()

for supplier in suppliers:

print("supplier\_name: {}\nexpected\_delivery\_date: {}\nactual\_delivery\_date: {}\nsupply\_name: {}\n".format(

supplier[0], supplier[1], supplier[2], supplier[3]))

# Insert values into deliveries table

del1 = "INSERT INTO deliveries (supplier\_id, inventory) VALUE (1, 50)"

del2 = "INSERT INTO deliveries (supplier\_id, inventory) VALUE (2, 40)"

del3 = "INSERT INTO deliveries (supplier\_id, inventory) VALUE (3, 60)"

myCursor.execute(del1)

myCursor.execute(del2)

myCursor.execute(del3)

mydb.commit()

# Print out values

print("-Displaying deliveries Records-")

query5 = "SELECT supplier\_id, inventory FROM deliveries"

myCursor.execute(query5)

deliveries = myCursor.fetchall()

for delivery in deliveries:

print("Supplier ID: {}\nInventory: {}\n".format(delivery[0], delivery[1]))

# Insert values into supplies table

spl1 = "INSERT INTO supplies (supply\_name, inventory) VALUE ('Corks', 50)"

spl2 = "INSERT INTO supplies (supply\_name, inventory) VALUE ('Boxes', 40)"

myCursor.execute(spl1)

myCursor.execute(spl2)

mydb.commit()

# Print out values

print("-Displaying Supplies Records-")

query6 = "SELECT supply\_name, inventory FROM supplies"

myCursor.execute(query6)

supplies = myCursor.fetchall()

for supply in supplies:

print("Supply Name: {}\nInventory: {}\n".format(supply[0], supply[1]))

# Insert values into employees table

emp1 = "INSERT INTO employees (first\_name, last\_name, job\_title) VALUE ('Stan', 'Bacchus', 'Co-Owner')"

emp2 = "INSERT INTO employees (first\_name, last\_name, job\_title) VALUE ('Davis', 'Bacchus', 'Co-Owner')"

emp3 = "INSERT INTO employees (first\_name, last\_name, job\_title) VALUE ('Janet', 'Collins', 'Finances and Payroll')"

emp4 = "INSERT INTO employees (first\_name, last\_name, job\_title) VALUE ('Roz', 'Murphy', 'Marketing Department')"

emp5 = "INSERT INTO employees (first\_name, last\_name, job\_title) VALUE ('Henry', 'Doyle', 'Production Line')"

emp6 = "INSERT INTO employees (first\_name, last\_name, job\_title) VALUE ('Maria', 'Costanza', 'Distribution')"

myCursor.execute(emp1)

myCursor.execute(emp2)

myCursor.execute(emp3)

myCursor.execute(emp4)

myCursor.execute(emp5)

myCursor.execute(emp6)

mydb.commit()

# Print out values

print("-Displaying Employee Records-")

query7 = "SELECT first\_name, last\_name, job\_title FROM employees"

myCursor.execute(query7)

employees = myCursor.fetchall()

for employee in employees:

print("First Name: {}\nLast Name: {}\nJob Title: {}\n".format(employee[0], employee[1],

employee[2]))

# Insert values into time sheet table

time1 = "INSERT INTO time\_sheet (week\_id, hours\_worked\_weekly, ot\_worked\_weekly, employee\_id) VALUE (1,45, 5, 1)"

time2 = "INSERT INTO time\_sheet (week\_id, hours\_worked\_weekly, ot\_worked\_weekly, employee\_id) VALUE (2, 50, 10, 2)"

time3 = "INSERT INTO time\_sheet (week\_id, hours\_worked\_weekly, ot\_worked\_weekly, employee\_id) VALUE (3, 32, 0, 3)"

time4 = "INSERT INTO time\_sheet (week\_id, hours\_worked\_weekly, ot\_worked\_weekly, employee\_id) VALUE (4, 45, 5, 4)"

time5 = "INSERT INTO time\_sheet (week\_id, hours\_worked\_weekly, ot\_worked\_weekly, employee\_id) VALUE (5, 60, 20, 5)"

time6 = "INSERT INTO time\_sheet (week\_id, hours\_worked\_weekly, ot\_worked\_weekly, employee\_id) VALUE (6, 40, 0, 6)"

myCursor.execute(time1)

myCursor.execute(time2)

myCursor.execute(time3)

myCursor.execute(time4)

myCursor.execute(time5)

myCursor.execute(time6)

mydb.commit()

# Print out values

print("-Displaying Time Sheet Records-")

query8 = "SELECT week\_id, hours\_worked\_weekly, ot\_worked\_weekly, employee\_id FROM time\_sheet"

myCursor.execute(query8)

time\_sheet = myCursor.fetchall()

for time in time\_sheet:

print("Week ID: {}\nHours worked weekly: {}\nOT hours worked weekly: {}\nemployee ID: {}\n".format(time[0], time[1],

time[2],

time[3]))

