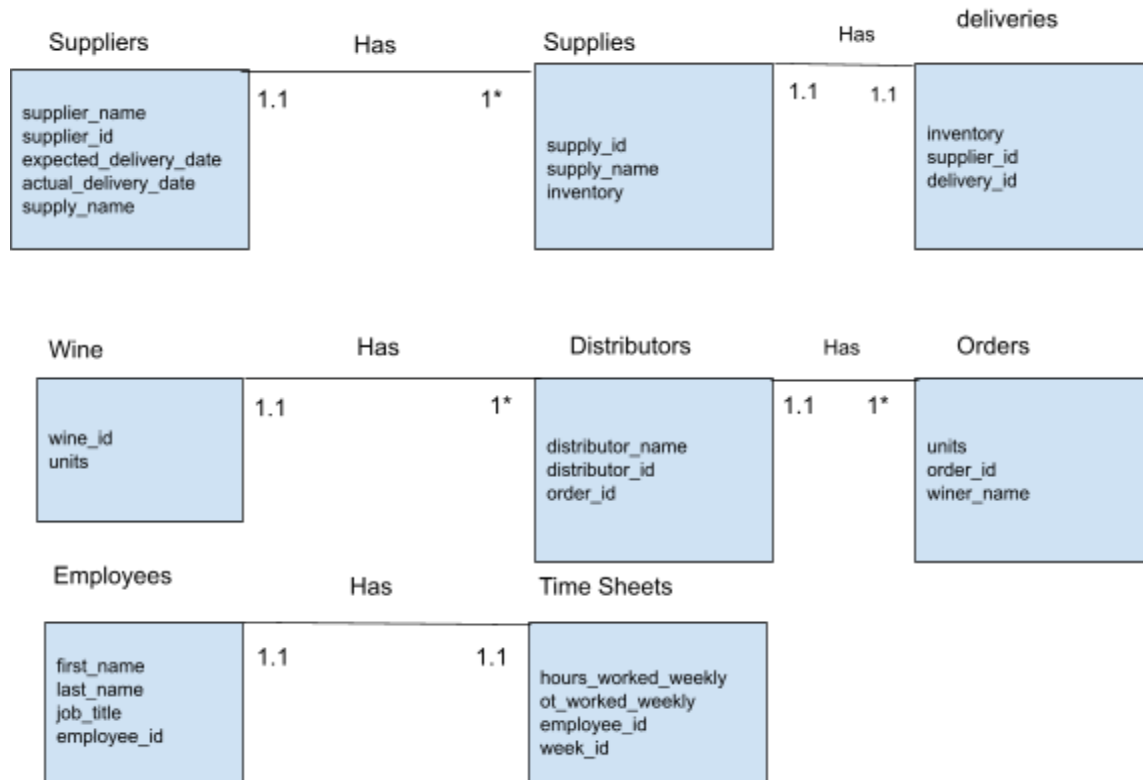


Group 2
Bacchus Winery Case Study

Revised ERD



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: {}".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: {}".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: {}".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]))
```

```
+-----+
| Tables_in_bacchus_winery |
+-----+
| deliveries
| distributors
| employees
| orders
| suppliers
| supplies
| time_sheet
| wine
+-----+
8 rows in set (0.01 sec)

mysql> █
```

Run: case_study_tables x

Creating table time_sheet: already exists.

-Displaying Employee Records-

First Name: Stan
Last Name: Bacchus
Job Title: Co-Owner

First Name: Davis
Last Name: Bacchus
Job Title: Co-Owner

First Name: Janet
Last Name: Collins
Job Title: Finances and Payroll

First Name: Roz
Last Name: Murphy
Job Title: Marketing Department

First Name: Henry
Last Name: Doyle
Job Title: Production Line

First Name: Maria
Last Name: Costanza
Job Title: Distribution

Process finished with exit code 0

employee_id	first_name	last_name	job_title
22	Roz	Murphy	Marketing Department
23	Henry	Doyle	Production Line
24	Maria	Costanza	Distribution

24 rows in set (0.00 sec)

```
mysql> SELECT * FROM employees;
```

employee_id	first_name	last_name	job_title
1	Stan	Bacchus	Co-Owner
2	Davis	Bacchus	Co-Owner
3	Janet	Collins	Finances and Payroll
4	Roz	Murphy	Marketing Department
5	Henry	Doyle	Production Line
6	Maria	Costanza	Distribution

6 rows in set (0.00 sec)

```
mysql> █
```

```

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)

```

Run: case_study_tables x

Creating table wine: OK

-Displaying Wine Records-

Wine Name: Merlot
Number of units: 50
Batch Month: January

Wine Name: Cabernet
Number of units: 40
Batch Month: February

Wine Name: Chablis
Number of units: 60
Batch Month: April

Wine Name: Chardonnay
Number of units: 50
Batch Month: March

clinty2710 — mysql -u root -p — 80x24

```

+-----+
| deliveries |
| distributors |
| employees |
| orders |
| suppliers |
| supplies |
| time_sheet |
| wine |
+-----+
8 rows in set (0.02 sec)

mysql> SELECT * FROM wine;
+-----+
| wine_id | wine_name | units | batch_month |
+-----+
| 1 | Merlot | 50 | January |
| 2 | Cabernet | 40 | February |
| 3 | Chablis | 60 | April |
| 4 | Chardonnay | 50 | March |
+-----+
4 rows in set (0.01 sec)

mysql>

```

```

41 " supply_name varchar(25) NOT NULL)")
42
43 # Insert values into Suppliers table
44 sup1 = "INSERT INTO suppliers (supplier_name, expected_delivery_date, actual_delivery_date, supply_name) \
45      "VALUE ('The New Corker', 20220110, 20220115, 'Corks')\"
46 sup2 = "INSERT INTO suppliers (supplier_name, expected_delivery_date, actual_delivery_date, supply_name) \
47      "VALUE ('Boxes n Stuff', 20220215, 20220215, 'Boxes')\"
48 sup3 = "INSERT INTO suppliers (supplier_name, expected_delivery_date, actual_delivery_date, supply_name) \
49      "VALUE ('Wine Supply', 20220320, 20220415, 'Tubing')\"
50 myCursor.execute(sup1)

```

Run: case_study_tables x

-Displaying Suppliers Records-

supplier_name: The New Corker
expected_delivery_date: 2022-01-10
actual_delivery_date: 2022-01-15
supply_name: Corks

supplier_name: Boxes n Stuff
expected_delivery_date: 2022-02-15
actual_delivery_date: 2022-02-15
supply_name: Boxes

supplier_name: Wine Supply
expected_delivery_date: 2022-03-20
actual_delivery_date: 2022-04-15
supply_name: Tubing

clinty2710 — mysql -u root -p — 101x23

```

mysql> SELECT * FROM wine;
+-----+
| wine_id | wine_name | units | batch_month |
+-----+
| 1 | Merlot | 50 | January |
| 2 | Cabernet | 40 | February |
| 3 | Chablis | 60 | April |
| 4 | Chardonnay | 50 | March |
+-----+
4 rows in set (0.01 sec)

mysql> SELECT * FROM suppliers;
+-----+
| supplier_id | supplier_name | expected_delivery_date | actual_delivery_date | supply_name |
+-----+
| 1 | The New Corker | 2022-01-10 | 2022-01-15 | Corks |
| 2 | Boxes n Stuff | 2022-02-15 | 2022-02-15 | Boxes |
| 3 | Wine Supply | 2022-03-20 | 2022-04-15 | Tubing |
+-----+
3 rows in set (0.01 sec)

mysql>

```

```
41 " supply_name varchar(25) NOT NULL)"
42
43 # Insert values into Suppliers table
44 sup1 = "INSERT INTO suppliers (supplier_name, expected_delivery_date, actual_delivery_date, supply_name) \
45 "VALUE ('The New Corker', 20220110, 20220115, 'Corks')\"
46 sup2 = "INSERT INTO suppliers (supplier_name, expected_delivery_date, actual_delivery_date, supply_name) \
47 "VALUE ('Boxes n Stuff', 20220215, 20220215, 'Boxes')\"
48 sup3 = "INSERT INTO suppliers (supplier_name, expected_delivery_date, actual_delivery_date, supply_name) \
49 "VALUE ('Wine Supply', 20220320, 20220415, 'Tubing')\"
50 myCursor.execute(sup1)
51 myCursor.execute(sup2)
52 myCursor.execute(sup3)
53 mydb.commit()
```

Run: case_study_tables

-Displaying Suppliers Records-

supplier_name: The New Corker
expected_delivery_date: 2022-01-10
actual_delivery_date: 2022-01-15
supply_name: Corks

supplier_name: Boxes n Stuff
expected_delivery_date: 2022-02-15
actual_delivery_date: 2022-02-15
supply_name: Boxes

supplier_name: Wine Supply
expected_delivery_date: 2022-03-20
actual_delivery_date: 2022-04-15
supply_name: Tubing

```
clinty2710 — mysql -u root -p — 101x13
4 rows in set (0.01 sec)
mysql> SELECT * FROM suppliers;
+-----+-----+-----+-----+-----+
| supplier_id | supplier_name | expected_delivery_date | actual_delivery_date | supply_name |
+-----+-----+-----+-----+-----+
| 1 | The New Corker | 2022-01-10 | 2022-01-15 | Corks |
| 2 | Boxes n Stuff | 2022-02-15 | 2022-02-15 | Boxes |
| 3 | Wine Supply | 2022-03-20 | 2022-04-15 | Tubing |
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
mysql>
```

```
# 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()

Run: case_study_tables
Press any key to continue...
```

-Displaying deliveries Records-

Supplier ID: 1
Inventory: 50

Supplier ID: 2
Inventory: 40

Supplier ID: 3
Inventory: 60

```
clinty2710 — mysql -u root -p — 101x13
mysql> SELECT * FROM deliveries;
+-----+-----+-----+
| delivery_id | supplier_id | inventory |
+-----+-----+-----+
| 1 | 1 | 50 |
| 2 | 2 | 40 |
| 3 | 3 | 60 |
+-----+-----+-----+
3 rows in set (0.00 sec)
mysql>
```

case_study_tables.py

```

57 time1 = "INSERT INTO time_sheet (week_id, hours_worked_weekly, ot_worked_weekly, employee_id) VALUE (1, 45, 5, 1)"
58 time2 = "INSERT INTO time_sheet (week_id, hours_worked_weekly, ot_worked_weekly, employee_id) VALUE (2, 50, 10, 2)"
59 time3 = "INSERT INTO time_sheet (week_id, hours_worked_weekly, ot_worked_weekly, employee_id) VALUE (3, 32, 0, 3)"
60 time4 = "INSERT INTO time_sheet (week_id, hours_worked_weekly, ot_worked_weekly, employee_id) VALUE (4, 45, 5, 4)"
61 time5 = "INSERT INTO time_sheet (week_id, hours_worked_weekly, ot_worked_weekly, employee_id) VALUE (5, 60, 20, 5)"
62 time6 = "INSERT INTO time_sheet (week_id, hours_worked_weekly, ot_worked_weekly, employee_id) VALUE (6, 40, 0, 6)"
63 myCursor.execute(time1)
for time in time_sheet

```

Run: case_study_tables

-Displaying Time Sheet Records-

Week ID: 1
Hours worked weekly: 45
OT hours worked weekly: 5
employee ID: 1

Week ID: 2
Hours worked weekly: 50
OT hours worked weekly: 10
employee ID: 2

Week ID: 3
Hours worked weekly: 32
OT hours worked weekly: 0
employee ID: 3

Week ID: 4
Hours worked weekly: 45
OT hours worked weekly: 5
employee ID: 4

clinty2710 — mysql -u root -p — 101x11

week_id	hours_worked_weekly	ot_worked_weekly	employee_id
1	45	5	1
2	50	10	2
3	32	0	3
4	45	5	4

4 rows in set (0.01 sec)

mysql>

case_study_tables.py

```

62 dis1 = "INSERT INTO distributors (distributor_name, order_id) VALUE ('Wines to GO', 2)"
63 dis2 = "INSERT INTO distributors (distributor_name, order_id) VALUE ('Partners in Wine', 3)"
64 dis3 = "INSERT INTO distributors (distributor_name, order_id) VALUE ('Sip Happens Co.', 1)"
65 myCursor.execute(dis1)
66 myCursor.execute(dis2)
67 myCursor.execute(dis3)
68 mydb.commit()
69
70 print("-Displaying Distributors-")
71 query3 = "SELECT distributor_name, order_id FROM distributors"
72 myCursor.execute(query3)
73 distributors = myCursor.fetchall()
74 for distributor in distributors:
75     print("Distributor Name: {}\nOrder ID: {}".format(distributor_name, order_id))

```

Run: case_studies

Creating table distributors: already exists.

-Displaying Distributors-

Distributor Name: Wines to GO
Order ID: 2

Distributor Name: Partners in Wine
Order ID: 3

Distributor Name: Sip Happens Co.
Order ID: 1

Process finished with exit code 0

clinty2710 — mysql -u root -p — 101x11

mysql> SELECT * FROM distributors;

distributor_id	distributor_name	order_id
1	Wines to GO	2
2	Partners in Wine	3
3	Sip Happens Co.	1

3 rows in set (0.00 sec)

mysql>

The screenshot shows a Python IDE with two files: `case_study_tables.py` and `case_studies.py`. The `case_studies.py` file contains the following code:

```
62 spl1 = "INSERT INTO supplies (supply_id, supply_name, inventory) VALUE (3, 'Corks', 50)"
63 spl2 = "INSERT INTO supplies (supply_id, supply_name, inventory) VALUE (4, 'Boxes', 40)"
64 myCursor.execute(spl1)
65 myCursor.execute(spl2)
66 mydb.commit()
67
68 # Print out values
69 print("-Displaying Supplies Records-")
70 query6 = "SELECT supply_id, supply_name, inventory FROM supplies"
71 myCursor.execute(query6)
```

The IDE's Run window shows the output of the script:

```
Creating table supplies: already exists.
-Displaying Supplies Records-
Supply ID: 1
Supply Name: Corks
Inventory: 50

Supply ID: 2
Supply Name: Boxes
Inventory: 40

Supply ID: 3
Supply Name: Corks
Inventory: 50

Supply ID: 4
Supply Name: Boxes
Inventory: 40
```

A MySQL terminal window titled `clinty2710 — mysql -u root -p — 101x11` displays the following table:

supply_id	supply_name	inventory
1	Corks	50
2	Boxes	40
3	Corks	50
4	Boxes	40

4 rows in set (0.01 sec)

The screenshot shows a Python IDE with two files: `case_study_tables.py` and `case_studies.py`. The `case_studies.py` file contains the following code:

```
63 orders1 = "INSERT INTO orders (units, wine_name) VALUE (10, 'Merlot')"
64 orders2 = "INSERT INTO orders (units, wine_name) VALUE (15, 'Cabernet')"
65 orders3 = "INSERT INTO orders (units, wine_name) VALUE (12, 'Chablis')"
66 orders4 = "INSERT INTO orders (units, wine_name) VALUE (10, 'Chardonnay')"
67 myCursor.execute(orders1)
68 myCursor.execute(orders2)
69 myCursor.execute(orders3)
70 myCursor.execute(orders4)
71 mydb.commit()
72
73 print("-Displaying Orders-")
74 query2 = "SELECT units, wine_name FROM orders"
```

The IDE's Run window shows the output of the script:

```
Creating table orders: already exists.
-Displaying Orders-
Number of units: 10
Wine Name: Merlot

Number of units: 15
Wine Name: Cabernet

Number of units: 12
Wine Name: Chablis

Number of units: 10
Wine Name: Chardonnay
```

A MySQL terminal window titled `clinty2710 — mysql -u root -p — 101x11` displays the following table:

order_id	units	wine_name
1	10	Merlot
2	15	Cabernet
3	12	Chablis
4	10	Chardonnay

4 rows in set (0.00 sec)