**Week 7 Module 10.1 Case Study Milestone #2**

Joel Atkinson, Zachary Baker, Kyle Klausen, Juan Macias Vasquez, Brittaney Perry-Morgan

Bellevue University

CSD310-H323 Database Development and Use (2255-DD)

**Adam Bailey**

July 13th, 2025

**Module 10.1 Case Study Milestone #2**

**Group Name:**

Group 1 The best group!

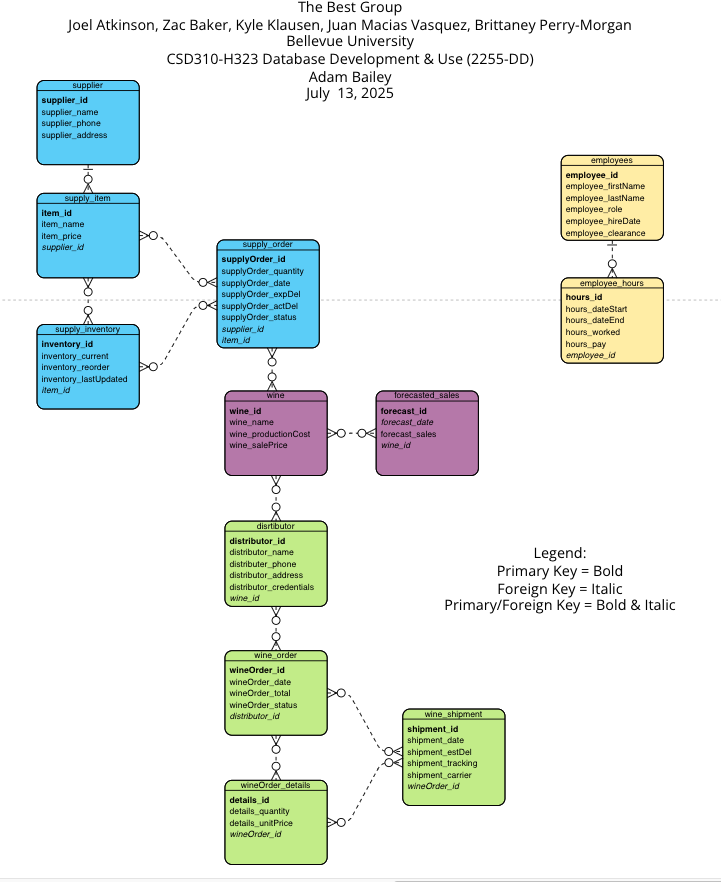
**Team Members:**

Joel Atkinson, Zachary Baker, Kyle Klausen, Juan Macias Vasquez, Brittaney Perry-Morgan

**Chosen Case Study:**

Bacchus Winery

Updates ERD



Python Code

#Team Members: Joel Atkinson, Zachary Baker, Kyle Klausen, Juan Macias Vasquez, Brittaney Perry-Morgan

# Date 07/13/2025

# Module 10.2 Group1 Winery Database

# Grouptables.py

import mysql.connector

# Connection configuration

config = {

'user': 'root',

'password': 'Powerful@2025', #Change to your own password

'host': 'localhost'

}

def execute\_sql\_file(cursor, filename):

with open(filename, 'r') as file:

lines = file.readlines()

statement = ''

for line in lines:

stripped = line.strip()

if not stripped or stripped.startswith('--') or stripped.startswith('#'):

continue

statement += ' ' + stripped

if stripped.endswith(';'):

try:

# Optional fix for subqueries with multiple results

if "SELECT" in statement.upper() and "FROM" in statement.upper():

if "employee\_id FROM employees WHERE employee\_firstName" in statement:

# Auto-fix: add LIMIT 1 to subqueries that cause errors

statement = statement.replace("')", "' LIMIT 1)") # crude but works here

cursor.execute(statement)

except mysql.connector.Error as err:

print("Error executing SQL:")

print(statement)

print("MySQL Error:", err)

statement = ''

def get\_table\_names(cursor):

cursor.execute("SHOW TABLES")

return [row[0] for row in cursor.fetchall()]

def get\_primary\_key(cursor, table):

cursor.execute(f"SHOW KEYS FROM {table} WHERE Key\_name = 'PRIMARY'")

result = cursor.fetchone()

return result[4] if result else None

def delete\_duplicates(cursor, table, primary\_key):

cursor.execute(f"SHOW COLUMNS FROM {table}")

columns = [row[0] for row in cursor.fetchall()]

non\_pk\_columns = [col for col in columns if col != primary\_key]

if not non\_pk\_columns:

return

match\_conditions = " AND ".join([f"t1.{col} = t2.{col}" for col in non\_pk\_columns])

sql = f"""

DELETE t1 FROM {table} t1

INNER JOIN {table} t2

WHERE t1.{primary\_key} > t2.{primary\_key}

AND {match\_conditions}

"""

try:

cursor.execute(sql)

except mysql.connector.Error as err:

print(f"Error deduplicating {table}: {err}")

def display\_table(cursor, table):

cursor.execute(f"SELECT \* FROM {table}")

rows = cursor.fetchall()

columns = [desc[0] for desc in cursor.description]

print(f"\n--- {table.upper()} ---")

print(" | ".join(columns))

print("-" \* 60)

for row in rows:

print(" | ".join(str(val) for val in row))

print(f"Total Rows: {len(rows)}")

def main():

conn = None

cursor = None

try:

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

cursor = conn.cursor()

# Ensure database exists and select it

cursor.execute("CREATE DATABASE IF NOT EXISTS Winery")

cursor.execute("USE Winery")

# Execute all SQL statements from file

execute\_sql\_file(cursor, 'tables.sql')

conn.commit()

# Confirm use of Winery

cursor.execute("USE Winery")

# Deduplicate and show tables

tables = get\_table\_names(cursor)

for table in tables:

pk = get\_primary\_key(cursor, table)

if pk:

delete\_duplicates(cursor, table, pk)

conn.commit()

for table in tables:

display\_table(cursor, table)

except mysql.connector.Error as err:

print(f"Database error: {err}")

finally:

if cursor:

cursor.close()

if conn and conn.is\_connected():

conn.close()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Screen shots of code running

