12/03/2022

**Group Members::** Ives, Aguilar, Pedersen, Crandall, Gayle

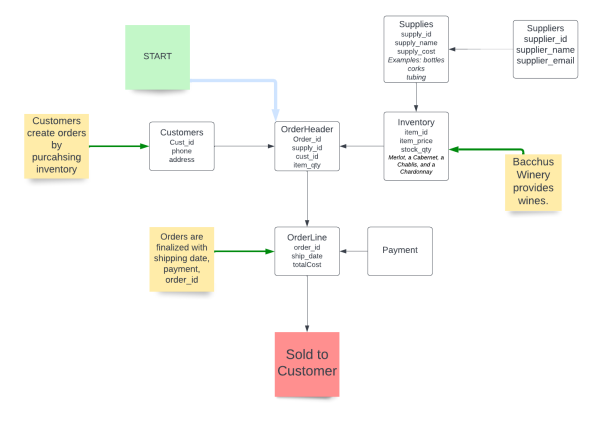
**Case Study:** Bacchus Winery

**Milestone #1:**

**Business Rules:**

* *An inbound order has a supplier.*
* *Inventory is increased by the number of inbound orders depending on the Supply\_no.*
* *Inventory has all its data like pricing and name stored inside the Item table.*
* *When bottles of wine are made internally, inventory would be tracked to reduce the number of supplied parts like bottles, corks, etc, and increase wine stock.*
* *Distributors create Outbound\_orders that can have many Items.*
* *Inventory orders (inbound\_orders) are tracked by comparing expected\_delivery\_date with actual\_delivery\_date*
* *Contact table houses employee, distributors, and suppliers contact information by distinct contact\_id’s*
* *One employee is associated with one payroll*
* *One department has many employees and managers*
* *One employee is associated with many work\_hours entries*

**Initial ERD:**



**Milestone #2:**

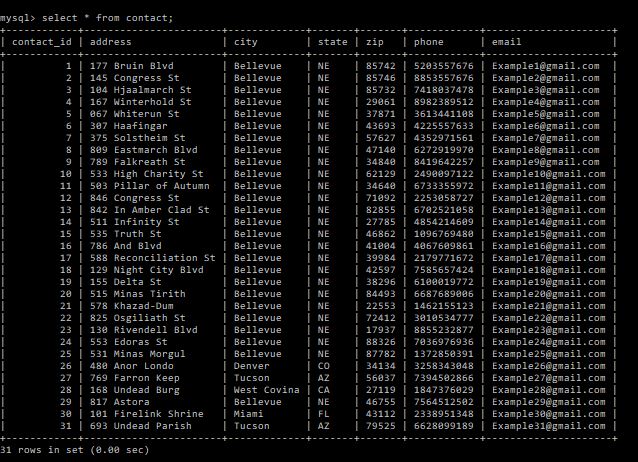
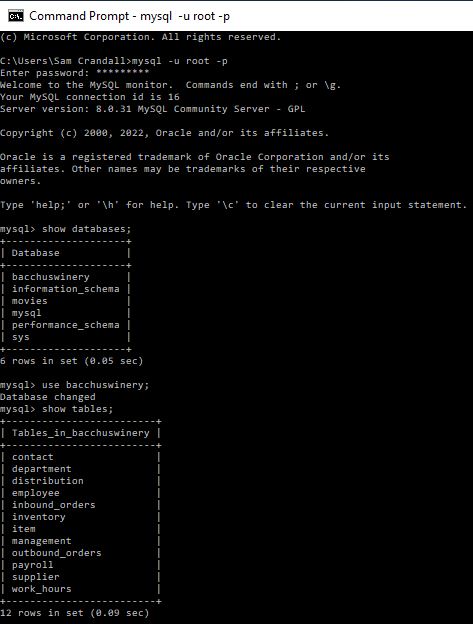
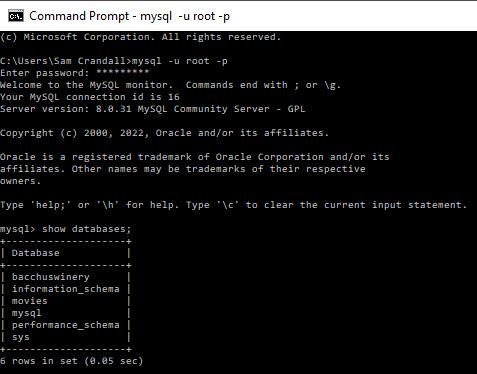
**Create Tables File(.sql):** *Bacchus\_Winery\_Table\_Inserts.sql*

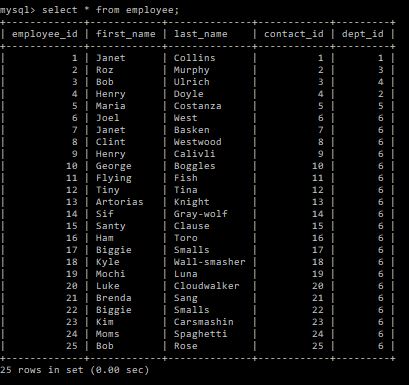
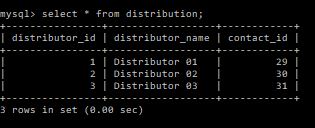
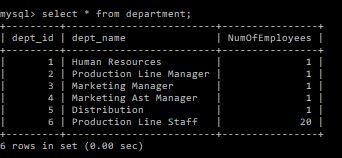
| */\*  Title: Bacchus\_Winery\_Table\_Inserts.sql  Author:TEAM INDIGO  Date: 12/04/22  Description: This .sql file creates a database titled bacchuswinery then creates 12 tables. \*/  create database bacchuswinery; USE bacchuswinery;  -- drop database user if exists  DROP USER IF EXISTS 'team\_indigo'@'localhost';  -- create team\_indigo and grant them all privileges to the bacchuswinery database  CREATE USER 'team\_indigo'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'pokemon';  -- grant all privileges to the `bacchuswinery` database to user team\_indigo on localhost  GRANT ALL PRIVILEGES ON `bacchuswinery`.\* TO 'team\_indigo'@'localhost';   DROP TABLE IF EXISTS Inbound\_orders; DROP TABLE IF EXISTS Outbound\_orders; DROP TABLE IF EXISTS Management; DROP TABLE IF EXISTS Payroll; DROP TABLE IF EXISTS Work\_Hours; DROP TABLE IF EXISTS Employee; DROP TABLE IF EXISTS Distribution; DROP TABLE IF EXISTS Inventory; DROP TABLE IF EXISTS Item; DROP TABLE IF EXISTS Department; DROP TABLE IF EXISTS Supplier; DROP TABLE IF EXISTS Contact;   CREATE TABLE `Inventory` (  `supply\_no` INT NOT NULL,  `item\_no` INT NOT NULL,  `inventory\_qty` INT NOT NULL,  PRIMARY KEY (`supply\_no`) );  CREATE TABLE `Item` (  `item\_no` INT NOT NULL,  `item\_name` VARCHAR(20) NOT NULL,  `item\_price` DECIMAL(6,2) NOT NULL,  PRIMARY KEY (`item\_no`) );  CREATE TABLE `Supplier` (  `supplier\_id` INT NOT NULL,  `supplier\_name` VARCHAR(20) NOT NULL,  `contact\_id` INT NOT NULL,  PRIMARY KEY (`supplier\_id`) );  CREATE TABLE `Inbound\_orders` (  `inventory\_order\_id` INT NOT NULL,  `supplier\_id` INT NOT NULL,  `expected\_delivery\_dt` DATE,  `actual\_delivery\_dt` DATE,  `supply\_no` INT NOT NULL,  `quantity` INT NOT NULL,  PRIMARY KEY (`inventory\_order\_id`) );  CREATE TABLE `Outbound\_orders` (  `order\_no` INT NOT NULL,  `item\_count` INT NOT NULL,  `total\_cost` DECIMAL(6,2) NOT NULL,  `item\_no` INT NOT NULL,  `distributor\_id` INT NOT NULL,  `order\_date` DATE NOT NULL,  PRIMARY KEY (`order\_no`) );  CREATE TABLE `Distribution` (  `distributor\_id` INT NOT NULL,  `distributor\_name` VARCHAR(20) NOT NULL,  `contact\_id` INT NOT NULL,  PRIMARY KEY (`distributor\_id`) );  CREATE TABLE `Contact` (  `contact\_id` INT NOT NULL,  `address` VARCHAR(50),  `city` VARCHAR(20),  `state` VARCHAR(20),  `zip` VARCHAR(20),  `phone` VARCHAR(20),  `email` VARCHAR(20),  PRIMARY KEY (`contact\_id`) );  CREATE TABLE `Department` (  `dept\_id` INT NOT NULL,  `dept\_name` VARCHAR(50),  `NumOfEmployees` INT,  PRIMARY KEY (`dept\_id`) );  CREATE TABLE `Payroll` (  `check\_no` INT NOT NULL,  `pay\_amount` DECIMAL(6,2) NOT NULL,  `pay\_date` DATE NOT NULL,  `employee\_id` INT NOT NULL,  PRIMARY KEY (`check\_no`) );  CREATE TABLE `Employee` (  `employee\_id` INT NOT NULL,  `first\_name` VARCHAR(20) NOT NULL,  `last\_name` VARCHAR(20) NOT NULL,  `contact\_id` INT NOT NULL,  `dept\_id` INT NOT NULL,  PRIMARY KEY (`employee\_id`) );  CREATE TABLE `Management` (  `mngmt\_id` INT NOT NULL,  `employee\_id` INT NOT NULL,  `dept\_id` INT NOT NULL,  `start\_date` DATE,  `end\_date` DATE,  PRIMARY KEY (`mngmt\_id`) );  CREATE TABLE `Work\_hours` (  `hours\_YTD` DECIMAL(9) NOT NULL,  `current\_week` INT NOT NULL,  `employee\_id` INT NOT NULL,  PRIMARY KEY (`employee\_id`) );  ALTER TABLE Inventory ADD CONSTRAINT FK\_inventory FOREIGN KEY (item\_no) REFERENCES Item(item\_no);  ALTER TABLE Supplier ADD CONSTRAINT FK\_supplier FOREIGN KEY (contact\_id) REFERENCES Contact(contact\_id);  ALTER TABLE Inbound\_orders ADD CONSTRAINT FK\_inb\_supplier FOREIGN KEY (supplier\_id) REFERENCES Supplier(supplier\_id);  ALTER TABLE Inbound\_orders ADD CONSTRAINT FK\_inb\_supply\_no FOREIGN KEY (supply\_no) REFERENCES Inventory(supply\_no);  ALTER TABLE Outbound\_orders ADD CONSTRAINT FK\_otb\_distributor FOREIGN KEY (distributor\_id) REFERENCES Distribution(distributor\_id);  ALTER TABLE Outbound\_orders ADD CONSTRAINT FK\_otb\_item FOREIGN KEY (item\_no) REFERENCES Item(item\_no);  ALTER TABLE Distribution ADD CONSTRAINT FK\_dist\_contact FOREIGN KEY (contact\_id) REFERENCES Contact(contact\_id);  ALTER TABLE Management ADD CONSTRAINT FK\_mgmt\_dpt FOREIGN KEY (dept\_id) REFERENCES Department(dept\_id);  ALTER TABLE Management ADD CONSTRAINT FK\_mgmt\_emp FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id);  ALTER TABLE Employee ADD CONSTRAINT FK\_emp\_contact FOREIGN KEY (contact\_id) REFERENCES Contact(contact\_id);  ALTER TABLE Employee ADD CONSTRAINT FK\_emp\_dept FOREIGN KEY (dept\_id) REFERENCES Department(dept\_id);  ALTER TABLE Payroll ADD CONSTRAINT FK\_pay\_emp FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id);  ALTER TABLE Work\_hours ADD CONSTRAINT FK\_work\_emp FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id);* |
| --- |

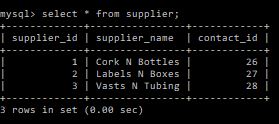
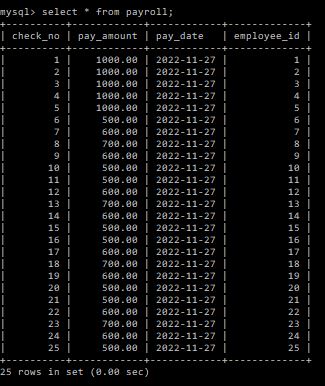
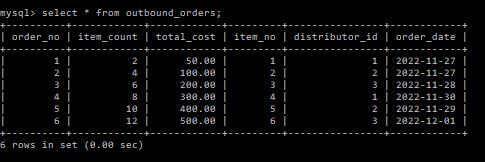
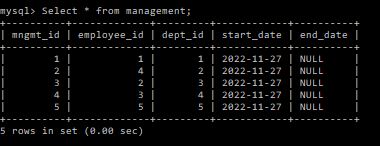
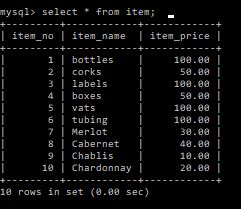
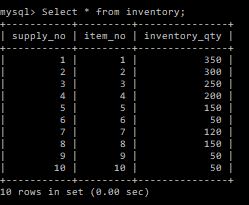
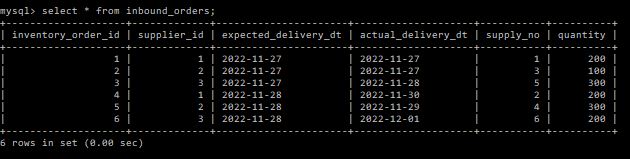
**Insert/Display Tables File(.py):** *bacchus\_winery\_inserts.py*

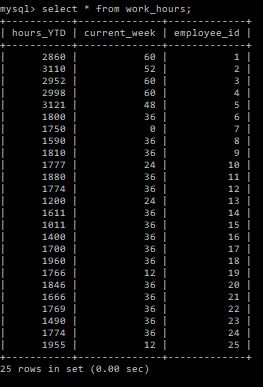
| **""" Team Indigo 12/4/2022 CSD 310, Milestone 2 Insert .py file Description: This .py file inserts data into the tables created by the file Bacchus\_Winery\_Table\_Inserts.sql After the data has been successfully inserted into the 12 tables, the 12 tables values are then displayed. """  import mysql.connector from mysql.connector import errorcode  config = {  "user": "\_\_\_\_\_\_\_",  "password": "\_\_\_\_\_\_\_!",  "host": "127.0.0.1",  "database": "BacchusWinery",  "raise\_on\_warnings": True } try:  db = 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")   cursor = db.cursor()   """ ----------------------------------------------- Display Tables ----------------------------------------------"""    def show\_contacts():  query = "SELECT contact\_id, address, city, email, phone, state, zip from contact"  cursor.execute(query)  contacts = cursor.fetchall()  for contact in contacts:  print("Contact ID: ", contact[0])  print("Address: ", contact[1])  print("City: ", contact[2])  print("Email: ", contact[3])  print("Phone: ", contact[4])  print("State: ", contact[5])  print("Zip: ", contact[6])  print(" ")    def show\_employees():  query = "SELECT employee\_id, first\_name, last\_name from employee"  cursor.execute(query)  employees = cursor.fetchall()  for employee in employees:  print("Employee\_ID: ", employee[0])  print("First Name: ", employee[1])  print("Last Name: ", employee[2])  print(" ")    def show\_management():  query = "SELECT mngmt\_id, dept\_id, employee\_id, end\_date, start\_date from management"  cursor.execute(query)  managers = cursor.fetchall()  for manager in managers:  print("Management ID: ", manager[0])  print("Department ID: ", manager[1])  print("Employee ID: ", manager[2])  print("End Date: ", manager[2])  print("Start Date: ", manager[2])  print(" ")    def show\_work\_hours():  query = "SELECT employee\_id, current\_week, hours\_YTD from work\_hours"  cursor.execute(query)  hours = cursor.fetchall()  for hour in hours:  print("Employee\_ID: ", hour[0])  print("Hours Worked (Current): ", hour[1])  print("Hours Worked (YTD): ", hour[2])  print(" ")    def show\_department():  query = "SELECT dept\_id, dept\_name, NumOfEmployees from department"  cursor.execute(query)  departments = cursor.fetchall()  for department in departments:  print("Department\_ID: ", department[0])  print("Department Name: ", department[1])  print("Number of Employees: ", department[2])  print(" ")    def show\_payroll():  query = "SELECT check\_no, pay\_amount, pay\_date, employee\_id from payroll"  cursor.execute(query)  payrolls = cursor.fetchall()  for payroll in payrolls:  print("Check Number: ", payroll[0])  print("Pay Amount: ", payroll[1])  print("Pay Date: ", payroll[2])  print("Employee ID: ", payroll[3])  print(" ")    def show\_inventory():  query = "SELECT supply\_no, item\_no, inventory\_qty from inventory"  cursor.execute(query)  inventories = cursor.fetchall()  for inventory in inventories:  print("Supply Number: ", inventory[0])  print("Item Number: ", inventory[1])  print("Inventory Quantity: ", inventory[2])  print(" ")    def show\_items():  query = "SELECT item\_no, item\_name, item\_price from item"  cursor.execute(query)  items = cursor.fetchall()  for item in items:  print("Item Number: ", item[0])  print("Item Name: ", item[1])  print("Item Price: ", item[2])  print(" ")    def show\_supplier():  query = "SELECT supplier\_id, supplier\_name, contact\_id from supplier"  cursor.execute(query)  suppliers = cursor.fetchall()  for supplier in suppliers:  print("Supplier ID: ", supplier[0])  print("Supplier Name: ", supplier[1])  print("Contact ID: ", supplier[2])  print(" ")    def show\_inbound\_orders():  query = "SELECT inventory\_order\_id, supplier\_id, expected\_delivery\_dt, actual\_delivery\_dt, supply\_no, " \  "quantity from inbound\_orders "  cursor.execute(query)  inbound\_orders = cursor.fetchall()  for inbound\_order in inbound\_orders:  print("Inventory Order ID: ", inbound\_order[0])  print("Supplier ID: ", inbound\_order[1])  print("Expected Delivery Date: ", inbound\_order[2])  print("Actual Delivery Date: ", inbound\_order[3])  print("Supply Number: ", inbound\_order[4])  print("Quantity: ", inbound\_order[5])  print(" ")    def show\_distribution():  query = "SELECT distributor\_id, contact\_id, distributor\_name from distribution"  cursor.execute(query)  distributors = cursor.fetchall()  for distributor in distributors:  print("Distributor ID: ", distributor[0])  print("Contact ID: ", distributor[1])  print("Distributor Name: ", distributor[2])  print(" ")    def show\_outbound\_orders():  query = "SELECT order\_no, item\_count, total\_cost, order\_date, distributor\_id, item\_no from outbound\_orders"  cursor.execute(query)  outbound\_orders = cursor.fetchall()  for outbound\_order in outbound\_orders:  print("Order Number: ", outbound\_order[0])  print("Item Count: ", outbound\_order[1])  print("Total Cost: ", outbound\_order[2])  print("Order Date: ", outbound\_order[3])  print("Distributor ID: ", outbound\_order[4])  print("Item Number: ", outbound\_order[5])  print(" ")    """ ----------------------------------------------- Add Inserts ----------------------------------------------"""   # CONTACT INSERT  contact\_insert\_statement = (  "INSERT INTO contact(contact\_id, address, city, email, phone, state, zip)" "VALUES (%s, %s, %s, "  "%s, %s, %s, %s) ")  contact\_list = [  # there are 5 managers, lets assume the winery is in Bellevue  ('1', '177 Bruin Blvd', 'Bellevue', 'Example1@gmail.com', '5203557676', 'NE', '85742'),  ('2', '145 Congress St', 'Bellevue', 'Example2@gmail.com', '8853557676', 'NE', '85746'),  ('3', '104 Hjaalmarch St', 'Bellevue', 'Example3@gmail.com', '7418037478', 'NE', '85732'),  ('4', '167 Winterhold St', 'Bellevue', 'Example4@gmail.com', '8982389512', 'NE', '29061'),  ('5', '067 Whiterun St', 'Bellevue', 'Example5@gmail.com', '3613441108', 'NE', '37871'),   # there are 20 employees under Henry Doyle, lets assume the winery is in Bellevue  ('6', '307 Haafingar', 'Bellevue', 'Example6@gmail.com', '4225557633', 'NE', '43693'),  ('7', '375 Solstheim St', 'Bellevue', 'Example7@gmail.com', '4352971561', 'NE', '57627'),  ('8', '809 Eastmarch Blvd', 'Bellevue', 'Example8@gmail.com', '6272919970', 'NE', '47140'),  ('9', '789 Falkreath St', 'Bellevue', 'Example9@gmail.com', '8419642257', 'NE', '34840'),  ('10', '533 High Charity St', 'Bellevue', 'Example10@gmail.com', '2490097122', 'NE', '62129'),  ('11', '503 Pillar of Autumn', 'Bellevue', 'Example11@gmail.com', '6733355972', 'NE', '34640'),  ('12', '846 Congress St', 'Bellevue', 'Example12@gmail.com', '2253058727', 'NE', '71092'),  ('13', '842 In Amber Clad St', 'Bellevue', 'Example13@gmail.com', '6702521058', 'NE', '82855'),  ('14', '511 Infinity St', 'Bellevue', 'Example14@gmail.com', '4854214609', 'NE', '27785'),  ('15', '535 Truth St', 'Bellevue', 'Example15@gmail.com', '1096769480', 'NE', '46862'),  ('16', '786 And Blvd', 'Bellevue', 'Example16@gmail.com', '4067609861', 'NE', '41004'),  ('17', '588 Reconciliation St', 'Bellevue', 'Example17@gmail.com', '2179771672', 'NE', '39984'),  ('18', '129 Night City Blvd', 'Bellevue', 'Example18@gmail.com', '7585657424', 'NE', '42597'),  ('19', '155 Delta St', 'Bellevue', 'Example19@gmail.com', '6100019772', 'NE', '38296'),  ('20', '515 Minas Tirith', 'Bellevue', 'Example20@gmail.com', '6687689006', 'NE', '84493'),  ('21', '578 Khazad-Dum', 'Bellevue', 'Example21@gmail.com', '1462155123', 'NE', '22553'),  ('22', '825 Osgiliath St', 'Bellevue', 'Example22@gmail.com', '3010534777', 'NE', '72412'),  ('23', '130 Rivendell Blvd', 'Bellevue', 'Example23@gmail.com', '8855232877', 'NE', '17937'),  ('24', '553 Edoras St', 'Bellevue', 'Example24@gmail.com', '7036976936', 'NE', '88326'),  ('25', '531 Minas Morgul', 'Bellevue', 'Example25@gmail.com', '1372850391', 'NE', '87782'),  # 3 different suppliers, so not in bellevue  ('26', '480 Anor Londo ', 'Denver', 'Example26@gmail.com', '3258343048', 'CO', '34134'),  ('27', '769 Farron Keep', 'Tucson', 'Example27@gmail.com', '7394502866', 'AZ', '56037'),  ('28', '168 Undead Burg', 'West Covina', 'Example28@gmail.com', '1847376029', 'CA', '27119'),  # distributor amount is not specified so lets just create 3  ('29', '817 Astora', 'Bellevue', 'Example29@gmail.com', '7564512502', 'NE', '46755'),  ('30', '101 Firelink Shrine', 'Miami', 'Example30@gmail.com', '2338951348', 'FL', '43112'),  ('31', '693 Undead Parish', 'Tucson', 'Example31@gmail.com', '6628099189', 'AZ', '79525')  ]   cursor.executemany(contact\_insert\_statement, contact\_list)  db.commit()   # Supplier INSERT  supplier\_insert\_statement = (  "INSERT INTO supplier(supplier\_id, supplier\_name, contact\_id)" "VALUES (%s, %s, %s)")  supplier\_list = [  ('1', 'Cork N Bottles', '26'),  ('2', 'Labels N Boxes', '27'),  ('3', 'Vasts N Tubing', '28'),  ]  cursor.executemany(supplier\_insert\_statement, supplier\_list)  db.commit()   # Department INSERT  department\_insert\_statement = (  "INSERT INTO department(dept\_id, dept\_name, NumOfEmployees)" "VALUES (%s, %s, %s)")  department\_list = [  # 6 entries on department\_list  ('1', 'Human Resources', '1'), # employ Janet Collins, who oversees all finances and payroll  ('2', 'Production Line Manager', '1'), # Henry Doyle, who manages the production line  ('3', 'Marketing Manager', '1'), # Roz Murphy, who heads up the marketing department  ('4', 'Marketing Ast Manager', '1'), # she has one assistant, Bob Ulrich, working for her  ('5', 'Distribution', '1'), # Maria Costanza, who is in charge of distribution  ('6', 'Production Line Staff', '20'), # 20 employees  ]  cursor.executemany(department\_insert\_statement, department\_list)  db.commit()   # Item INSERT  item\_insert\_statement = (  "INSERT INTO item(item\_no, item\_name, item\_price)" "VALUES (%s, %s, %s)")  item\_list = [  ('1', 'bottles', '100'),  ('2', 'corks', '50'),  ('3', 'labels', '100'),  ('4', 'boxes', '50'),  ('5', 'vats', '100'),  ('6', 'tubing', '100'),  ('7', 'Merlot', '30'),  ('8', 'Cabernet', '40'),  ('9', 'Chablis', '10'),  ('10', 'Chardonnay', '20'),  ]  cursor.executemany(item\_insert\_statement, item\_list)  db.commit()   # Inventory INSERT  inventory\_insert\_statement = (  "INSERT INTO inventory(supply\_no, item\_no, inventory\_qty)" "VALUES (%s, %s, %s)")  inventory\_list = [  ('1', '1', '350'),  ('2', '2', '300'),  ('3', '3', '250'),  ('4', '4', '200'),  ('5', '5', '150'),  ('6', '6', '50'),  ('7', '7', '120'),  ('8', '8', '150'),  ('9', '9', '50'),  ('10', '10', '50'),  ]  cursor.executemany(inventory\_insert\_statement, inventory\_list)  db.commit()   # Distribution INSERT  distribution\_insert\_statement = (  "INSERT INTO distribution(distributor\_id, contact\_id, distributor\_name)" "VALUES (%s, %s, %s)")  distributor\_list = [  ('1', '29', 'Distributor 01'),  ('2', '30', 'Distributor 02'),  ('3', '31', 'Distributor 03'),  ]  cursor.executemany(distribution\_insert\_statement, distributor\_list)  db.commit()   # EMPLOYEE INSERT  employee\_insert\_statement = (  "INSERT INTO employee(employee\_id, first\_name, last\_name, contact\_id, dept\_id)"   "VALUES (%s, %s, %s, %s, %s)")  employee\_list = [  # there should be 25 Employees  # 5 managers  ('1', 'Janet', 'Collins', '1', '1'),  ('2', 'Roz', 'Murphy', '2', '3'),  ('3', 'Bob', 'Ulrich', '3', '4'),  ('4', 'Henry', 'Doyle', '4', '2'),  ('5', 'Maria', 'Costanza', '5', '5'),   # employees under Henry Doyle  ('6', 'Joel', 'West', '6', '6'),  ('7', 'Janet', 'Basken', '7', '6'),  ('8', 'Clint', 'Westwood', '8', '6'),  ('9', 'Henry', 'Calivli', '9', '6'),  ('10', 'George', 'Boggles', '10', '6'),  ('11', 'Flying', 'Fish', '11', '6'),  ('12', 'Tiny', 'Tina', '12', '6'),  ('13', 'Artorias', 'Knight', '13', '6'),  ('14', 'Sif', 'Gray-wolf', '14', '6'),  ('15', 'Santy', 'Clause', '15', '6'),  ('16', 'Ham', 'Toro', '16', '6'),  ('17', 'Biggie', 'Smalls', '17', '6'),  ('18', 'Kyle', 'Wall-smasher', '18', '6'),  ('19', 'Mochi', 'Luna', '19', '6'),  ('20', 'Luke', 'Cloudwalker', '20', '6'),  ('21', 'Brenda', 'Sang', '21', '6'),  ('22', 'Biggie', 'Smalls', '22', '6'),  ('23', 'Kim', 'Carsmashin', '23', '6'),  ('24', 'Moms', 'Spaghetti', '24', '6'),  ('25', 'Bob', 'Rose', '25', '6')   ]   cursor.executemany(employee\_insert\_statement, employee\_list)  db.commit()   # Work\_Hours INSERT  workhours\_insert\_statement = (  "INSERT INTO work\_hours(employee\_id, current\_week, hours\_YTD)" "VALUES (%s, %s, %s)")  work\_hours\_list = [  # there should be 25 Employees  # 5 managers  ('1', '60', '2860'),  ('2', '52', '3110'),  ('3', '60', '2952'),  ('4', '60', '2998'),  ('5', '48', '3121'),   # employees under Henry Doyle  ('6', '36', '1800'),  ('7', '0', '1750'),  ('8', '36', '1590'),  ('9', '36', '1810'),  ('10', '24', '1777'),  ('11', '36', '1880'),  ('12', '36', '1774'),  ('13', '24', '1200'),  ('14', '36', '1611'),  ('15', '36', '1011'),  ('16', '36', '1400'),  ('17', '36', '1700'),  ('18', '36', '1960'),  ('19', '12', '1766'),  ('20', '36', '1846'),  ('21', '36', '1666'),  ('22', '36', '1769'),  ('23', '36', '1490'),  ('24', '36', '1774'),  ('25', '12', '1955')  ]  cursor.executemany(workhours\_insert\_statement, work\_hours\_list)  db.commit()   # Payroll INSERT  payroll\_insert\_statement = (  "INSERT INTO payroll(check\_no, pay\_amount, pay\_date, employee\_id)" "VALUES (%s, %s, %s, %s)")  payroll\_list = [  # 5 managers  ('1', '1000.00', '2022-11-27', '1'),  ('2', '1000.00', '2022-11-27', '2'),  ('3', '1000.00', '2022-11-27', '3'),  ('4', '1000.00', '2022-11-27', '4'),  ('5', '1000.00', '2022-11-27', '5'),  # 20 Employees  ('6', '500.00', '2022-11-27', '6'),  ('7', '600.00', '2022-11-27', '7'),  ('8', '700.00', '2022-11-27', '8'),  ('9', '600.00', '2022-11-27', '9'),  ('10', '500.00', '2022-11-27', '10'),  ('11', '500.00', '2022-11-27', '11'),  ('12', '600.00', '2022-11-27', '12'),  ('13', '700.00', '2022-11-27', '13'),  ('14', '600.00', '2022-11-27', '14'),  ('15', '500.00', '2022-11-27', '15'),  ('16', '500.00', '2022-11-27', '16'),  ('17', '600.00', '2022-11-27', '17'),  ('18', '700.00', '2022-11-27', '18'),  ('19', '600.00', '2022-11-27', '19'),  ('20', '500.00', '2022-11-27', '20'),  ('21', '500.00', '2022-11-27', '21'),  ('22', '600.00', '2022-11-27', '22'),  ('23', '700.00', '2022-11-27', '23'),  ('24', '600.00', '2022-11-27', '24'),  ('25', '500.00', '2022-11-27', '25')  ]  cursor.executemany(payroll\_insert\_statement, payroll\_list)  db.commit()   # Inbound Orders INSERT  inbound\_orders\_insert\_statement = (  "INSERT INTO inbound\_orders(inventory\_order\_id, supplier\_id, expected\_delivery\_dt, actual\_delivery\_dt, "  "supply\_no, quantity)" "VALUES (%s, %s, %s, %s, %s, %s)")  inbound\_orders\_list = [  ('1', '1', '2022-11-27', '2022-11-27', '1', '200'),  ('2', '2', '2022-11-27', '2022-11-27', '3', '100'),  ('3', '3', '2022-11-27', '2022-11-28', '5', '300'),  ('4', '1', '2022-11-28', '2022-11-30', '2', '200'),  ('5', '2', '2022-11-28', '2022-11-29', '4', '300'),  ('6', '3', '2022-11-28', '2022-12-01', '6', '200'),  ]  cursor.executemany(inbound\_orders\_insert\_statement, inbound\_orders\_list)  db.commit()   # Outbound Orders INSERT  outbound\_orders\_insert\_statement = (  "INSERT INTO outbound\_orders(order\_no, item\_count, total\_cost, order\_date, distributor\_id, item\_no)" "VALUES "  "(%s, %s, %s, %s, %s, %s)")  outbound\_orders\_list = [  ('1', '2', '50', '2022-11-27', '1', '1'),  ('2', '4', '100', '2022-11-27', '2', '2'),  ('3', '6', '200', '2022-11-28', '3', '3'),  ('4', '8', '300', '2022-11-30', '1', '4'),  ('5', '10', '400', '2022-11-29', '2', '5'),  ('6', '12', '500', '2022-12-01', '3', '6'),  ]  cursor.executemany(outbound\_orders\_insert\_statement, outbound\_orders\_list)  db.commit()   # Management INSERT  management\_insert\_statement = (  "INSERT INTO management(mngmt\_id, dept\_id, employee\_id, start\_date)" "VALUES (%s, %s, %s, %s)")  management\_list = [  ('1', '1', '1', '2022-11-27'),  ('2', '2', '4', '2022-11-27'),  ('3', '3', '2', '2022-11-27'),  ('4', '4', '3', '2022-11-27'),  ('5', '5', '5', '2022-11-27'),  ]  cursor.executemany(management\_insert\_statement, management\_list)  db.commit()   # Display Output  print("-- Contacts --\n")  show\_contacts()  print("-- Employees --\n")  show\_employees()  print("-- Work Hours --\n")  show\_work\_hours()  print("-- Department --\n")  show\_department()  print("-- Payroll --\n")  show\_payroll()  print("-- Inventory --\n")  show\_inventory()  print("-- Items --\n")  show\_items()  print("-- Suppliers --\n")  show\_supplier()  print("-- Inbound Orders --\n")  show\_inbound\_orders()  print("-- Outbound Orders --\n")  show\_outbound\_orders()  print("-- Distribution --\n")  show\_distribution()  print("-- Management --\n")  show\_management()  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)  finally:  db.close()** |
| --- |

**Screenshots of Output:**

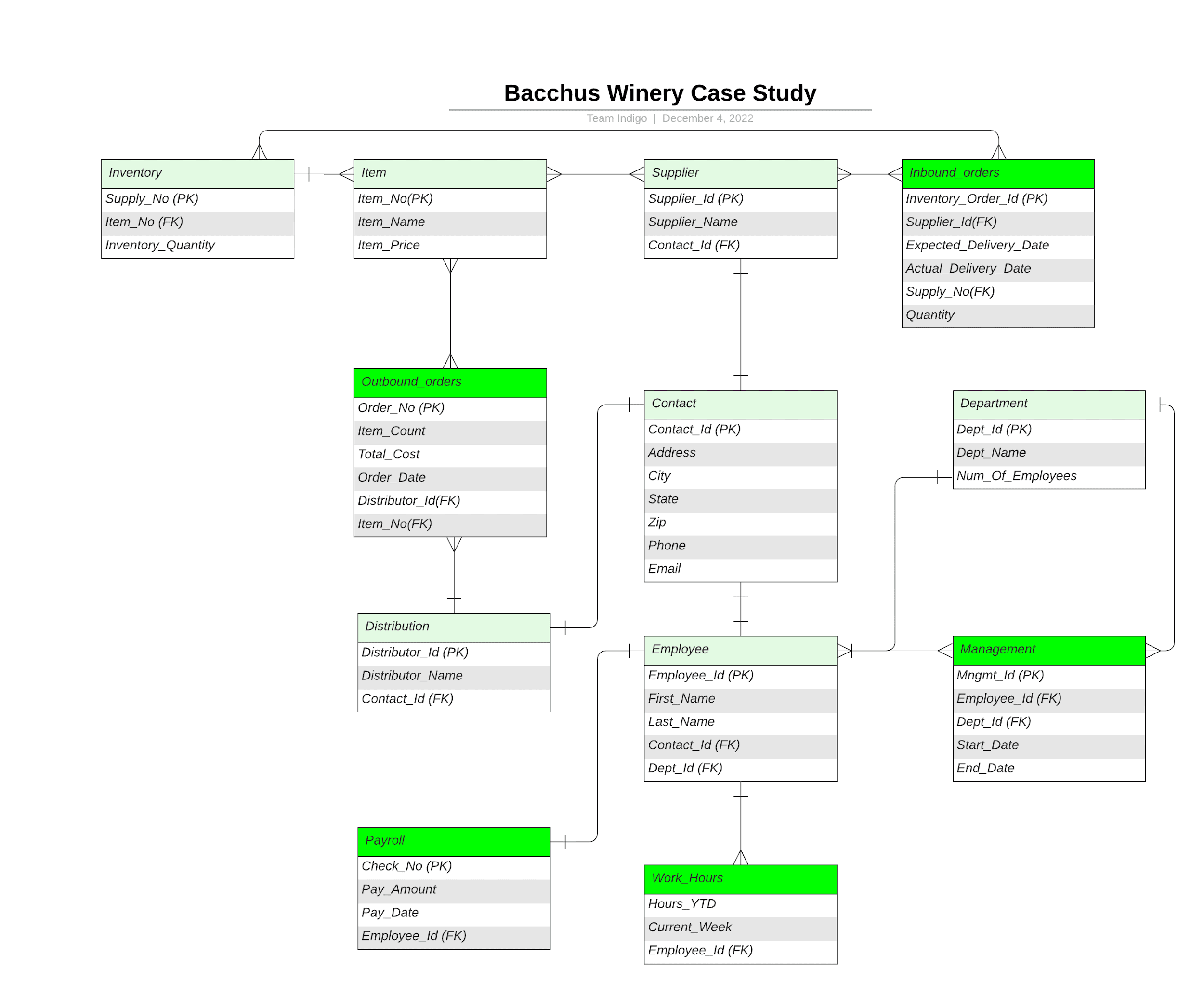
****

****

****

****

**Revised ERD:** Image Below

[****](https://lucid.app/lucidchart/ea6047ce-7a2b-4ffb-9b26-b5a914458b5a/edit?page=0&v=6337&s=544)