

Benjamin Z. Chen

CU ID: (4 letters - 4 digits): bech1695

GitHub ID: benzionchen

Number of hours it took to complete the lab: 20 hours

Overview

The Final Project will give you hands-on practice in designing, creating, loading and using a database that solves problems for a given company. From the description of the company and the forms that it uses to run its business, you will create the design of the database, implement that design, place sample data into those tables, and provide queries to get the data required by the company employees to run and plan their business.

You should implement your design using a MySQL database. If you wish to use another DBMS (either relational or NoSQL), you MUST obtain **written** approval from the instructor to use an alternative DBMS (such as PostgreSQL, SQLite, MongoDB, Cassandra, Amazon Dynamo, Redis, Elasticsearch, etc.)

Objectives

1. Familiarize yourself with an unfamiliar company via the Case Study
2. Gain an understanding of the unfamiliar company's data as presented in the Case Study
3. Determine the precise SCOPE of your proposed solution
4. Design a database to meet the needs of the client company
5. Create scripts (appropriate for the DBMS you have chosen) and then create the database you have designed and build the database
6. Create test data appropriate for the Case Study, and insert the data into the database you have created
7. Run queries against your database

Case Study Scenarios

For this project, you may choose to use one of TWO different case studies:

- Eden Landscaping – Tracking Inventory (same Case Study text from previous assignment)
- MyMobile Telecom – Customer Agreements & Calls (contained in this document)

1. Submit a Project Proposal (due Week 13)

- Specify the database management system being used and your selection of Case Study to complete.
- Describe the scope of your solution and details about the tables you will be creating.
- Include a document or drawing depicting your initial database design using the information in the Case Study as your input. Your design should include:
 - All entities (a person, place, thing or event about which you are keeping data) with proper keys defined
 - All attributes, by entity, with data type, length, and constraints defined as appropriate
 - If your design is relational, your database design (Data Model or ERD) must show all relationships between tables showing captions (1-way is OK), and proper optionality and cardinality
 - If your design is NOT relational, a drawing will suffice. You should show how the entities in your database are related to each other

2. Final Project Report (due Finals Week)

- Include updated Database Design information
 - Text files or screenshots of the scripts/commands you used to create and populate your database
 - Documentation (screen shots) demonstrating the physical implementation of your database that matches your design
 - The query code and the output of the queries against your database as described in detail requirements described below
- ANY CHANGES MADE TO PROPOSAL NEED TO BE EXPLICITLY STATED IN FINAL PROJECT SUBMISSION

Your results for this project assignment can be captured in a document (such as a .txt file, MS Word or similar tool.) Please then save your final deliverable document as a PDF for submission. The final PDF deliverable document you submit for this project must consist of four sections:

- The first section is a picture of your complete database design
- The second section is text files that contain the scripts you used to create the database
- The third section is text containing the scripts you used to populate the data in your database
- The fourth section contains your query code and the output of the queries against your database

Step-by-Step Instructions

1. Read the overview of the business case you have chosen
2. As you read the case study, document the ENTITIES and ATTRIBUTES you observe as you learn about how the organization operates. (Suggestion: use a spreadsheet like HW # 1)
3. As you read the case study, you must consider each **business process** that affects the in-scope business process. (Inventory for Eden, *Customer Plans and Transactions* for MyMobile)
4. Once you finish identifying entities and attributes, you can design the database that will become the foundation for the client queries in the next phase of this project.
5. Draw a design that includes all entities and attributes and the relationships among them. If designing a relational solution, all entities/attributes should be in third normal form.
6. Walk through the design and compare it to each business process to ensure that it contains all entities and attributes necessary to support the business processes described for your client.
7. Once the database design is complete, run necessary scripts to create the database.
8. Populate each entity in your database with several rows of test data of your own creation. Capture and turn in text files containing the scripts used to populate the database:
 1. For Eden Landscaping
 - Add some Plants in inventory (at least 8)
 - Add some Landscaping Materials in inventory (at least 6)
 - Add some Customers (at least 4)
 - Add some Suppliers (at least 2)
 - Add some Retail Sales Tickets and Details (at least 4 tickets with 4 items each)

9. Create and run queries to show the following:

1. For Eden Landscaping

- A list of Customers showing name, address, city, state, zip, phone1 and phone2
- A list of plants in inventory showing category (trees, shrubs, perennials), SKU, plant description, and size (where known.) Be sure to include the quantity currently in inventory.
- A list of materials in inventory showing category (gravel, mulch, stone, etc.), SKU, and description. Be sure to include the quantity currently in inventory based on the Unit of Measure for each material. Gravel, mulch, topsoil and sand are measured in cubic yards. Stone and pavers are by pallet. Timbers are per each, sold by size.

Eden Landscaping

You must study the Eden Landscaping Case Study. As you read the Case Study, you must pay close attention to every reference to the data that is collected and used by Eden Landscaping.

For this assignment, you must play the role of a consultant who has been hired by a client (Eden Landscaping) to design and create a database for them.

The SCOPE of this assignment includes your database design for ONLY the **inventory** kept by Eden for their **LIVE PLANT STOCK** and **LANDSCAPING MATERIALS**.

Your scope for this project excludes:

- The scope of this assignment specifically **EXCLUDES** the tracking of any items kept in inventory for the purpose of retail sales through Eden's **retail store** (like shovels, rakes, seeds, bags of fertilizer, hoses, etc.)
- The scope of the assignment **EXCLUDES** any inventory of tools and equipment used by Eden's crews as they go out to customers' locations and do landscaping jobs (like the backhoe, wheelbarrows, chain saws, shovels, etc.)

Your scope for this project **includes**:

- **Plants and landscaping materials** that are sold to customers who come to the store, make a purchase, and pick up and carry out their purchased items
- Plants and landscaping materials that are sold to customers who come to the store, make a purchase, and have Eden deliver their purchased items to their home
- Plants and landscaping materials that are sold to customers as part of a landscaping service contract ("job"), where Eden loads up a truck and sends a crew to the customer's home to do a landscaping project

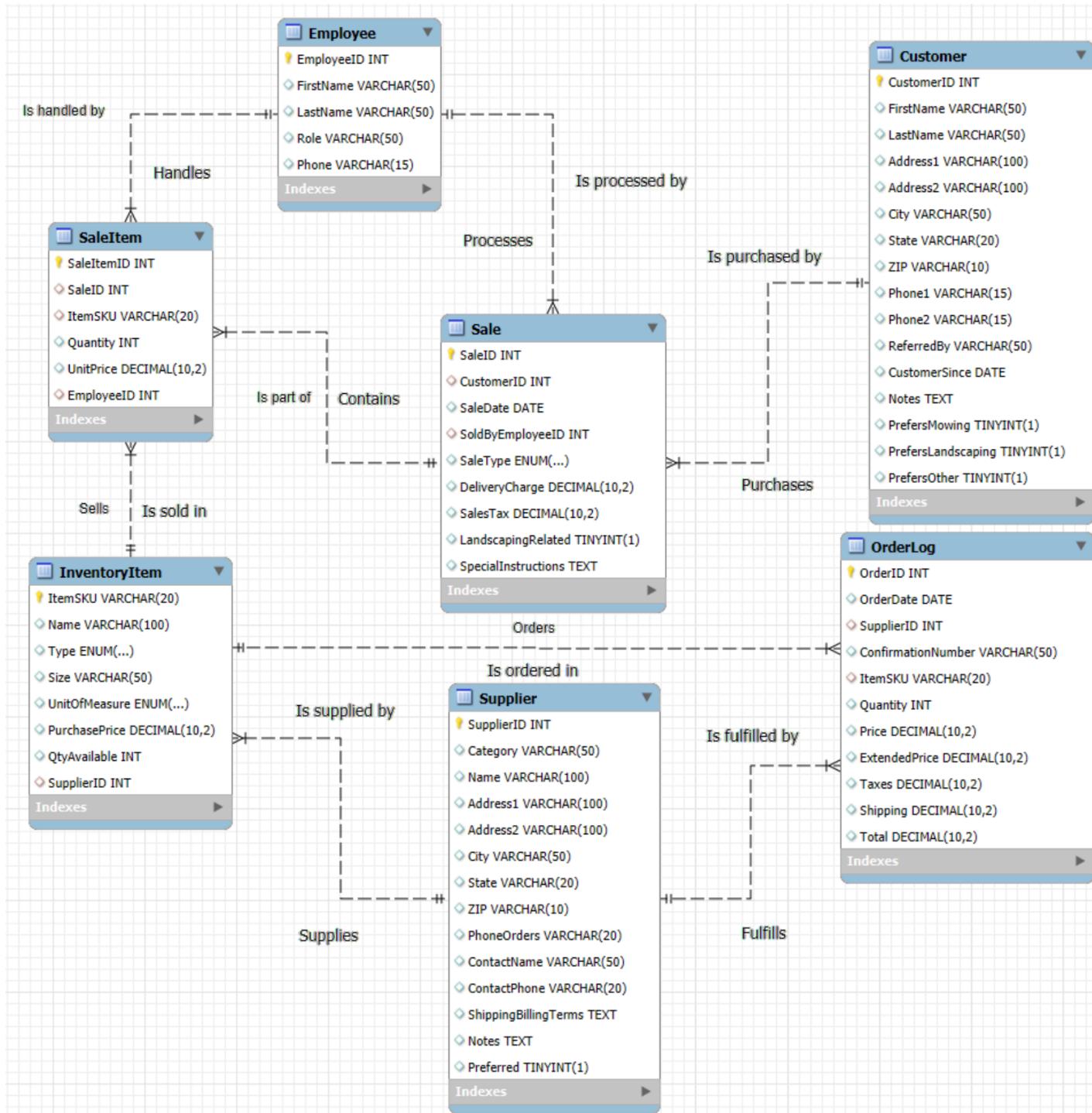
The case study includes in-depth descriptions of the business processes that center around the keeping of inventory, including business activities that **ADD** items into inventory (such as Orders and receiving Shipments) and the business activities that **SUBTRACT** items from inventory (such as sales, landscaping jobs, and deliveries of purchased items to customers.)

Your database design must support the following business processes at Eden:

- The ordering of plants and landscaping materials from suppliers
- Taking physical inventory to see what's in stock
- Receiving incoming shipments of plants and landscaping materials from suppliers
- Fulfilling customer orders and purchases, including landscaping jobs

Final Project

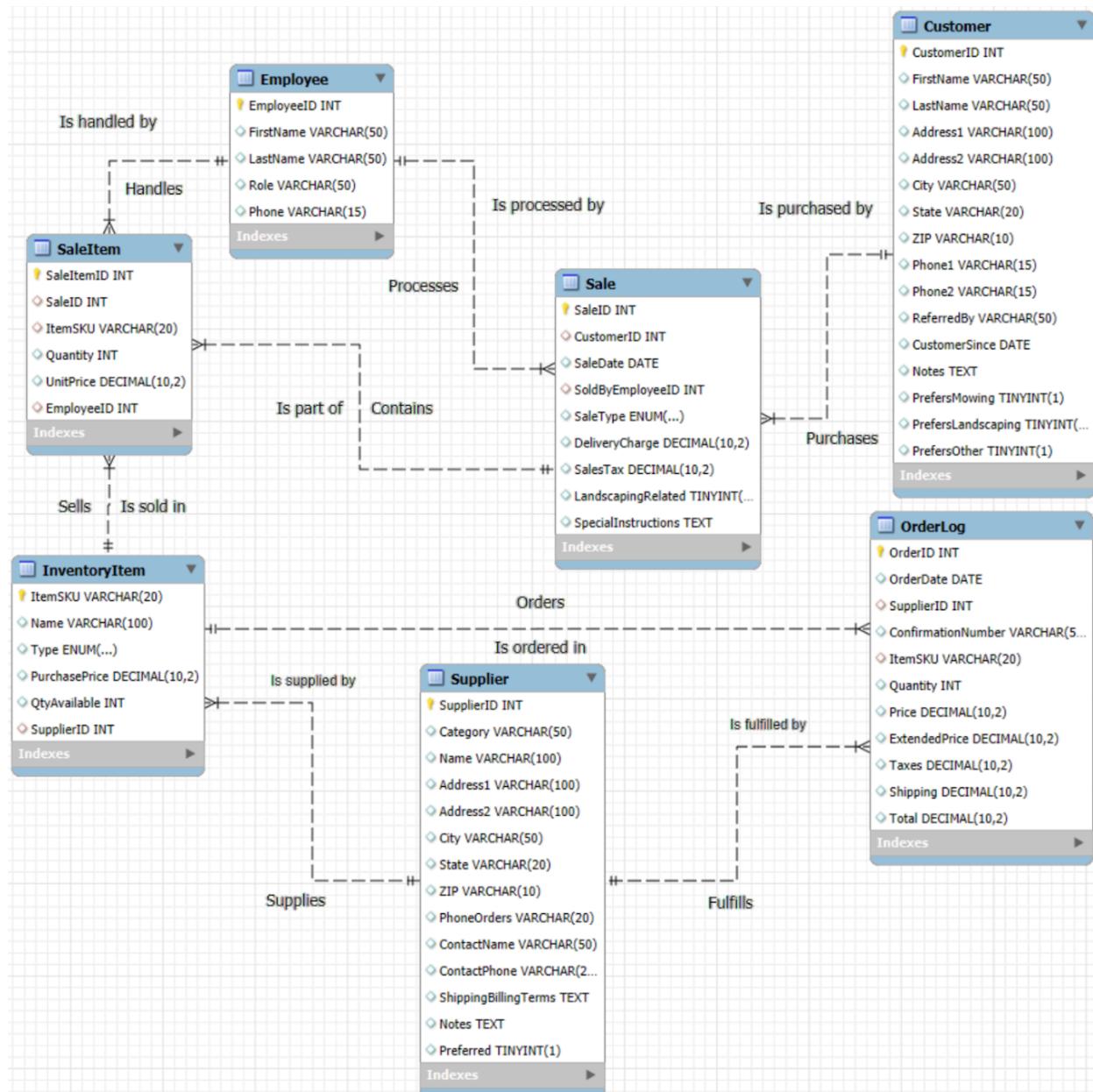
The first section is a picture of your complete database design (final draft ERD):



- I made some changes between this and my proposal

- InventoryItem now has Size VARCHAR(50) to accommodate for the way we're measuring the landscaping material (for example: '6\' tall' for live plants), giving data on maturity of the plant before customers buy it
 - I also added UnitOfMeasure ENUM(...) to account for "('Each', 'CubicYards', 'Pallet')" because the way we measured stones vs. gravel, etc. were different per the proposal's rules and I caught this mistake
- I decided to keep "PrefersMowing" and "PrefersLandscaping", etc. because though it's not central to our project scope, customers that buy products *can* still prefer things that are outside of the product they're purchasing (i.e. somebody buying gravel doesn't mean they don't prefer mowing services).

Submitted project proposal ERD:



The second section is text files that contain the scripts you used to create the database

```
1      -- DROP existing tables to start fresh
2 •  DROP TABLE IF EXISTS OrderLog;
3 •  DROP TABLE IF EXISTS SaleItem;
4 •  DROP TABLE IF EXISTS Sale;
5 •  DROP TABLE IF EXISTS InventoryItem;
6 •  DROP TABLE IF EXISTS Supplier;
7 •  DROP TABLE IF EXISTS Customer;
8 •  DROP TABLE IF EXISTS Employee;
9
10     -- Employee table: for sales handling
11 •  CREATE TABLE Employee (
12         EmployeeID INT PRIMARY KEY AUTO_INCREMENT,
13         FirstName VARCHAR(50),
14         LastName VARCHAR(50),
15         Role VARCHAR(50),
16         Phone VARCHAR(15)
17 );
18
19     -- Supplier table: who provides inventory
20 •  CREATE TABLE Supplier (
21         SupplierID INT PRIMARY KEY AUTO_INCREMENT,
22         Category VARCHAR(50),
23         Name VARCHAR(100),
24         Address1 VARCHAR(100),
25         Address2 VARCHAR(100),
26         City VARCHAR(50),
27         State VARCHAR(20),
28         ZIP VARCHAR(10),
29         PhoneOrders VARCHAR(20),
30         ContactName VARCHAR(50),
31         ContactPhone VARCHAR(20),
32         ShippingBillingTerms TEXT,
33         Notes TEXT,
34         Preferred TINYINT(1)
35 );
```

```
37      -- Inventory item table: plants/materials being stocked and sold
38 • CREATE TABLE InventoryItem (
39     ItemSKU VARCHAR(20) PRIMARY KEY,
40     Name VARCHAR(100),
41     Type ENUM(
42         'Tree', 'Shrub', 'Perennial',
43         'Mulch', 'Gravel', 'Topsoil',
44         'Stone', 'Paver', 'Timber', 'Sand'
45     ),
46     Size VARCHAR(50),
47     UnitOfMeasure ENUM('Each', 'CubicYards', 'Pallet'),
48     PurchasePrice DECIMAL(10,2),
49     QtyAvailable INT,
50     SupplierID INT,
51     FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)
52 );
53
54      -- Customer table
55 • CREATE TABLE Customer (
56     CustomerID INT PRIMARY KEY AUTO_INCREMENT,
57     FirstName VARCHAR(50),
58     LastName VARCHAR(50),
59     Address1 VARCHAR(100),
60     Address2 VARCHAR(100),
61     City VARCHAR(50),
62     State VARCHAR(20),
63     ZIP VARCHAR(10),
64     Phone1 VARCHAR(15),
65     Phone2 VARCHAR(15),
66     ReferredBy VARCHAR(50),
67     CustomerSince DATE,
68     Notes TEXT,
69     PrefersMowing TINYINT(1),
70     PrefersLandscaping TINYINT(1),
71     PrefersOther TINYINT(1)
72 );
```

```
74      -- Sale table: records of all transactions
75 • CREATE TABLE Sale (
76     SaleID INT PRIMARY KEY AUTO_INCREMENT,
77     CustomerID INT,
78     SaleDate DATE,
79     SoldByEmployeeID INT,
80     SaleType ENUM('InStore', 'Delivery', 'Landscaping'),
81     DeliveryCharge DECIMAL(10,2),
82     SalesTax DECIMAL(10,2),
83     LandscapingRelated TINYINT(1),
84     SpecialInstructions TEXT,
85     FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
86     FOREIGN KEY (SoldByEmployeeID) REFERENCES Employee(EmployeeID)
87 );
88
89      -- SaleItem table: details of items sold in each Sale
90 • CREATE TABLE SaleItem (
91     SaleItemID INT PRIMARY KEY AUTO_INCREMENT,
92     SaleID INT,
93     ItemSKU VARCHAR(20),
94     Quantity INT,
95     UnitPrice DECIMAL(10,2),
96     EmployeeID INT,
97     FOREIGN KEY (SaleID) REFERENCES Sale(SaleID),
98     FOREIGN KEY (ItemSKU) REFERENCES InventoryItem(ItemSKU),
99     FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
100 );
```

```

102      -- OrderLog table: adds inventory by recording supplier shipments
103 • CREATE TABLE OrderLog (
104      OrderID INT PRIMARY KEY AUTO_INCREMENT,
105      OrderDate DATE,
106      SupplierID INT,
107      ConfirmationNumber VARCHAR(50),
108      ItemSKU VARCHAR(20),
109      Quantity INT,
110      Price DECIMAL(10,2),
111      ExtendedPrice DECIMAL(10,2),
112      Taxes DECIMAL(10,2),
113      Shipping DECIMAL(10,2),
114      Total DECIMAL(10,2),
115      FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID),
116      FOREIGN KEY (ItemSKU) REFERENCES InventoryItem(ItemSKU)
117 );

```

The third section is text containing the scripts you used to populate the data in your database

```

1 -- 2 Suppliers
2 INSERT INTO Supplier (Category, Name, Address1, Address2, City, State, ZIP, PhoneOrders, ContactName, ContactPhone, ShippingBillingTerms, Notes, Preferred)
3 VALUES
4 ('Mulch', 'A.A. "Buck" Jones & Assoc, Inc', 'N/A', '', 'Grayson', 'GA', '34217', '770-963-8227', 'N/A', '770-963-8227', 'Net 30 days', 'Fax: 770-339-0899', 1),
5 ('Live Stock', 'Basler Tree Farm & Nursery', '2953 Yaeger Rd', '', 'Saint Louis', 'MO', '63129', '314-487-4698', 'N/A', '314-487-4698', 'Prepay', 'No Fax', 0),
6 ('Gravel', 'Dirt Cheap Landscape', '2900 N Highway 7', '', 'Hot Springs Village', 'AR', '71909', '501-623-5099', 'N/A', '501-623-5099', 'Net 15 days', 'Fax: 501-412-3562', 0),
7 ('Stone', 'Garick Corporation', 'N/A', '', 'Cleveland', 'OH', '59763', '216-581-0100', 'N/A', '216-581-0100', 'Prepay', 'Fax: 216-581-4712', 0);
8 -- I debated removing the attributes like "ContactName" from the table since we're not going to use it, but I kept it there because in the real world, it's there just in case

```

```

10    -- Inventory Items
11 • INSERT INTO InventoryItem (ItemSKU, Name, Type, Size, UnitOfMeasure, PurchasePrice, QtyAvailable, SupplierID)
12 VALUES
13    -- 8 Live Plant Stock
14    ('TREE001', 'Red Maple', 'Tree', '6\'', 'Each', 30.00, 10, 2),
15    ('TREE002', 'American Walnut', 'Tree', '8\'', 'Each', 35.00, 8, 2),
16    ('SHRB001', 'Azalea "Red Bells"', 'Shrub', '2\'', 'Each', 12.00, 15, 2),
17    ('SHRB002', 'Honeysuckle "Trumpet"', 'Shrub', '3\'', 'Each', 10.00, 20, 2),
18    ('PER001', 'Hydrangea "Masja"', 'Perennial', '2\'', 'Each', 8.00, 25, 2),
19    ('PER002', 'Rosa "Snowcarpet"', 'Perennial', '1.5\'', 'Each', 7.50, 30, 2),
20    ('TREE003', 'Silver Maple', 'Tree', '10\'', 'Each', 45.00, 5, 2),
21    ('PER003', 'Hydrangea arborescens', 'Perennial', '3\'', 'Each', 10.00, 12, 2),
22
23    -- 6 Landscaping Materials
24    ('MULCH01', 'Pine Bark, Med.', 'Mulch', NULL, 'CubicYards', 22.00, 40, 1),
25    ('MULCH02', 'Pine Bark, Fine', 'Mulch', NULL, 'CubicYards', 21.00, 30, 1),
26    ('GRVL01', 'Quaker Blend', 'Gravel', NULL, 'CubicYards', 32.00, 35, 3),
27    ('GRVL02', 'Marble Chips', 'Gravel', NULL, 'CubicYards', 33.00, 25, 3),
28    ('STONE01', 'Keystone', 'Stone', NULL, 'Pallet', 90.00, 18, 4),
29    ('STONE02', 'Natural Limestone', 'Stone', NULL, 'Pallet', 88.00, 15, 4);

31
32    -- 4 Customers w/ made up information
33
34 • INSERT INTO Customer (FirstName, LastName, Address1, Address2, City, State, ZIP, Phone1, Phone2, ReferredBy, CustomerSince, Notes, PrefersMowing, PrefersLandscaping, PrefersOther)
35 VALUES
36    ('John', 'Smith', '100 Maple St', '', 'Longmont', 'MO', '63030', '314-111-2222', '314-111-3333', 'Google', '2023-01-10', '', 1, 0, 0),
37    ('Sara', 'Johnson', '222 Oak Ave', '', 'Longmont', 'MO', '63030', '314-444-5555', '314-666-7777', 'Referral', '2023-03-05', '', 0, 1, 0),
38    ('Emily', 'Nguyen', '300 Elm Dr', 'Apt 4', 'Longmont', 'MO', '63030', '314-888-9999', '314-000-1111', '', '2024-02-20', '', 1, 1, 0),
39    ('Carlos', 'Martinez', '455 Pine Rd', '', 'Longmont', 'MO', '63030', '314-222-3333', '314-444-6666', 'Instagram', '2024-01-10', '', 0, 0, 1);
40
41    -- Employees related to sales (working the register)
42
43 • INSERT INTO Employee (FirstName, LastName, Role, Phone)
44 VALUES
45    ('Rachel', 'Marshall', 'Office Manager', '314-555-0001'),
46    ('Tom', 'Flynn', 'Retail Manager', '314-555-0002');

47
48    -- 4 Retail Sales Tickets
49
50 • INSERT INTO Sale (CustomerID, SaleDate, SoldByEmployeeID, SaleType, DeliveryCharge, SalesTax, LandscapingRelated, SpecialInstructions)
51 VALUES
52    (1, '2024-04-01', 1, 'InStore', 0.00, 2.45, 0, 'Requested smaller bags.'),
53    (2, '2024-04-02', 2, 'InStore', 0.00, 3.10, 0, 'Assisted by Rachel.'),
54    (3, '2024-04-03', 1, 'InStore', 0.00, 4.25, 0, 'Picked up in afternoon.'),
55    (4, '2024-04-04', 2, 'InStore', 0.00, 3.80, 0, 'Used loyalty card.');

```

```
53      -- Sale Items
54 •  INSERT INTO SaleItem (SaleID, ItemSKU, Quantity, UnitPrice, EmployeeID)
55      VALUES
56      -- Ticket 1
57      (1, 'TREE001', 1, 55.00, 1),
58      (1, 'SHRB001', 2, 30.00, 1),
59      (1, 'MULCH01', 1, 20.00, 1),
60      (1, 'GRVL01', 1, 30.00, 1),
61
62      -- Ticket 2
63      (2, 'TREE002', 1, 40.00, 2),
64      (2, 'GRVL01', 2, 32.00, 2),
65      (2, 'STONE01', 1, 90.00, 2),
66      (2, 'MULCH02', 1, 21.00, 2),
67
68      -- Ticket 3
69      (3, 'PER001', 3, 20.00, 1),
70      (3, 'TREE003', 1, 45.00, 2),
71      (3, 'MULCH01', 2, 22.00, 1),
72      (3, 'STONE02', 1, 88.00, 2),
73
74      -- Ticket 4
75      (4, 'TREE003', 1, 50.00, 2),
76      (4, 'PER002', 2, 25.00, 2),
77      (4, 'GRVL02', 1, 33.00, 2),
78      (4, 'STONE01', 1, 88.00, 1);
```

The fourth section contains your query code and the output of the queries against your database

Query customer data

```
1 •  SELECT
2      CONCAT(FirstName, ' ', LastName) AS FullName,
3      Address1,
4      Address2,
5      City,
6      State,
7      ZIP,
8      Phone1,
9      Phone2
10     FROM Customer;
```

	FullName	Address1	Address2	City	State	ZIP	Phone1	Phone2
▶	John Smith	100 Maple St		Longmont	MO	63030	314-111-2222	314-111-3333
	Sara Johnson	222 Oak Ave		Longmont	MO	63030	314-444-5555	314-666-7777
	Emily Nguyen	300 Elm Dr	Apt 4	Longmont	MO	63030	314-888-9999	314-000-1111
	Carlos Martinez	455 Pine Rd		Longmont	MO	63030	314-222-3333	314-444-6666

Query plant data

```
1 •  SELECT
2      Type AS Category,
3      ItemSKU,
4      Name AS Description,
5      Size,
6      QtyAvailable
7      FROM InventoryItem
8      WHERE Type IN ('Tree', 'Shrub', 'Perennial');
```

Result Grid				
Edit:				
Category	ItemSKU	Description	Size	QtyAvailable
Perennial	PER001	Hydrangea "Masja"	2'	25
Perennial	PER002	Rosa "Snowcarpet"	1.5'	30
Perennial	PER003	Hydrangea arborescens	3'	12
Shrub	SHRB001	Azalea "Red Bells"	2'	15
Shrub	SHRB002	Honeysuckle "Trumpet"	3'	20
Tree	TREE001	Red Maple	6'	10
Tree	TREE002	American Walnut	8'	8
Tree	TREE003	Silver Maple	10'	5
NULL	NULL	NULL	NULL	NULL

Query landscaping material data

```
1 •  SELECT
2      Type AS MaterialCategory,
3      ItemSKU,
4      Name AS Description,
5      UnitOfMeasure,
6      QtyAvailable
7  FROM InventoryItem
8 WHERE Type IN ('Mulch', 'Gravel', 'Topsoil', 'Stone', 'Paver', 'Timber', 'Sand');
```

	MaterialCategory	ItemSKU	Description	UnitOfMeasure	QtyAvailable
▶	Gravel	GRVL01	Quaker Blend	CubicYards	35
	Gravel	GRVL02	Marble Chips	CubicYards	25
	Mulch	MULCH01	Pine Bark, Med.	CubicYards	40
	Mulch	MULCH02	Pine Bark, Fine	CubicYards	30
	Stone	STONE01	Keystone	Pallet	18
	Stone	STONE02	Natural Limestone	Pallet	15
*	NULL	NULL	NULL	NULL	NULL

Query what material belongs to which supplier

```
1 •  SELECT
2      I.ItemSKU,
3      I.Name AS Description,
4      I.Type AS MaterialCategory,
5      I.SupplierID,
6      S.Name AS SupplierName
7  FROM InventoryItem I
8  JOIN Supplier S ON I.SupplierID = S.SupplierID
9 WHERE I.Type IN ('Mulch', 'Gravel', 'Topsoil', 'Stone', 'Paver', 'Timber', 'Sand');
```

	ItemSKU	Description	MaterialCategory	SupplierID	SupplierName
▶	MULCH01	Pine Bark, Med.	Mulch	1	A.A. "Buck" Jones & Assoc, Inc
	MULCH02	Pine Bark, Fine	Mulch	1	A.A. "Buck" Jones & Assoc, Inc
	GRVL01	Quaker Blend	Gravel	3	Dirt Cheap Landscape
	GRVL02	Marble Chips	Gravel	3	Dirt Cheap Landscape
	STONE01	Keystone	Stone	4	Garick Corporation
	STONE02	Natural Limestone	Stone	4	Garick Corporation

Raw SQL Code (if for any reason you have trouble reading the picture/screenshots)

For creating the database

```
-- DROP existing tables to start fresh
DROP TABLE IF EXISTS OrderLog;
DROP TABLE IF EXISTS SaleItem;
DROP TABLE IF EXISTS Sale;
DROP TABLE IF EXISTS InventoryItem;
DROP TABLE IF EXISTS Supplier;
DROP TABLE IF EXISTS Customer;
DROP TABLE IF EXISTS Employee;

-- Employee table: for sales handling
CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Role VARCHAR(50),
    Phone VARCHAR(15)
);

-- Supplier table: who provides inventory
CREATE TABLE Supplier (
    SupplierID INT PRIMARY KEY AUTO_INCREMENT,
    Category VARCHAR(50),
    Name VARCHAR(100),
    Address1 VARCHAR(100),
    Address2 VARCHAR(100),
    City VARCHAR(50),
    State VARCHAR(20),
    ZIP VARCHAR(10),
    PhoneOrders VARCHAR(20),
    ContactName VARCHAR(50),
    ContactPhone VARCHAR(20),
    ShippingBillingTerms TEXT,
    Notes TEXT,
    Preferred TINYINT(1)
);

-- Inventory item table: plants/materials being stocked and sold
CREATE TABLE InventoryItem (
    ItemSKU VARCHAR(20) PRIMARY KEY,
    Name VARCHAR(100),
    Type ENUM(
```

```

'Tree', 'Shrub', 'Perennial',
'Mulch', 'Gravel', 'Topsoil',
'Stone', 'Paver', 'Timber', 'Sand'
),
Size VARCHAR(50),
UnitOfMeasure ENUM('Each', 'CubicYards', 'Pallet'),
PurchasePrice DECIMAL(10,2),
QtyAvailable INT,
SupplierID INT,
FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)
);

```

-- Customer table

```

CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Address1 VARCHAR(100),
    Address2 VARCHAR(100),
    City VARCHAR(50),
    State VARCHAR(20),
    ZIP VARCHAR(10),
    Phone1 VARCHAR(15),
    Phone2 VARCHAR(15),
    ReferredBy VARCHAR(50),
    CustomerSince DATE,
    Notes TEXT,
    PrefersMowing TINYINT(1),
    PrefersLandscaping TINYINT(1),
    PrefersOther TINYINT(1)
);

```

-- Sale table: records of all transactions

```

CREATE TABLE Sale (
    SaleID INT PRIMARY KEY AUTO_INCREMENT,
    CustomerID INT,
    SaleDate DATE,
    SoldByEmployeeID INT,
    SaleType ENUM('InStore', 'Delivery', 'Landscaping'),
    DeliveryCharge DECIMAL(10,2),
    SalesTax DECIMAL(10,2),
    LandscapingRelated TINYINT(1),
    SpecialInstructions TEXT,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
);

```

```
FOREIGN KEY (SoldByEmployeeID) REFERENCES Employee(EmployeeID)
);
```

-- SaleItem table: details of items sold in each Sale

```
CREATE TABLE SaleItem (
    SaleItemID INT PRIMARY KEY AUTO_INCREMENT,
    SaleID INT,
    ItemSKU VARCHAR(20),
    Quantity INT,
    UnitPrice DECIMAL(10,2),
    EmployeeID INT,
    FOREIGN KEY (SaleID) REFERENCES Sale(SaleID),
    FOREIGN KEY (ItemSKU) REFERENCES InventoryItem(ItemSKU),
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);
```

-- OrderLog table: adds inventory by recording supplier shipments

```
CREATE TABLE OrderLog (
    OrderID INT PRIMARY KEY AUTO_INCREMENT,
    OrderDate DATE,
    SupplierID INT,
    ConfirmationNumber VARCHAR(50),
    ItemSKU VARCHAR(20),
    Quantity INT,
    Price DECIMAL(10,2),
    ExtendedPrice DECIMAL(10,2),
    Taxes DECIMAL(10,2),
    Shipping DECIMAL(10,2),
    Total DECIMAL(10,2),
    FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID),
    FOREIGN KEY (ItemSKU) REFERENCES InventoryItem(ItemSKU)
);
```

For populating the database

-- 2 Suppliers

```
INSERT INTO Supplier (Category, Name, Address1, Address2, City, State, ZIP, PhoneOrders,
ContactName, ContactPhone, ShippingBillingTerms, Notes, Preferred)
VALUES
('Mulch', 'A.A. "Buck" Jones & Assoc, Inc', 'N/A', "", 'Grayson', 'GA', '34217', '770-963-8227',
'N/A', '770-963-8227', 'Net 30 days', 'Fax: 770-339-0899', 1),
('Live Stock', 'Basler Tree Farm & Nursery', '2953 Yaeger Rd', "", 'Saint Louis', 'MO', '63129',
'314-487-4698', 'N/A', '314-487-4698', 'Prepay', 'No Fax', 0),
```

('Gravel', 'Dirt Cheap Landscape', '2900 N Highway 7', "", 'Hot Springs Village', 'AR', '71909',
'501-623-5099', 'N/A', '501-623-5099', 'Net 15 days', 'Fax: 501-412-3562', 0),
(('Stone', 'Garick Corporation', 'N/A', "", 'Cleveland', 'OH', '59763', '216-581-0100', 'N/A',
'216-581-0100', 'Prepay', 'Fax: 216-581-4712', 0);
-- I debated removing the attributes like "ContactName" from the table since we're not going to
use it, but I kept it there because in the real world, it's there just in case

-- Inventory Items

INSERT INTO InventoryItem (ItemSKU, Name, Type, Size, UnitOfMeasure, PurchasePrice,
QtyAvailable, SupplierID)

VALUES

-- 8 Live Plant Stock

('TREE001', 'Red Maple', 'Tree', '6\'', 'Each', 30.00, 10, 2),
(('TREE002', 'American Walnut', 'Tree', '8\'', 'Each', 35.00, 8, 2),
(('SHRB001', 'Azalea "Red Bells"', 'Shrub', '2\'', 'Each', 12.00, 15, 2),
(('SHRB002', 'Honeysuckle "Trumpet"', 'Shrub', '3\'', 'Each', 10.00, 20, 2),
(('PER001', 'Hydrangea "Masja"', 'Perennial', '2\'', 'Each', 8.00, 25, 2),
(('PER002', 'Rosa "Snowcarpet"', 'Perennial', '1.5\'', 'Each', 7.50, 30, 2),
(('TREE003', 'Silver Maple', 'Tree', '10\'', 'Each', 45.00, 5, 2),
(('PER003', 'Hydrangea arborescens', 'Perennial', '3\'', 'Each', 10.00, 12, 2),

-- 6 Landscaping Materials

('MULCH01', 'Pine Bark, Med.', 'Mulch', NULL, 'CubicYards', 22.00, 40, 1),
(('MULCH02', 'Pine Bark, Fine', 'Mulch', NULL, 'CubicYards', 21.00, 30, 1),
(('GRVL01', 'Quaker Blend', 'Gravel', NULL, 'CubicYards', 32.00, 35, 3),
(('GRVL02', 'Marble Chips', 'Gravel', NULL, 'CubicYards', 33.00, 25, 3),
(('STONE01', 'Keystone', 'Stone', NULL, 'Pallet', 90.00, 18, 4),
(('STONE02', 'Natural Limestone', 'Stone', NULL, 'Pallet', 88.00, 15, 4);

-- 4 Customers w/ made up information

INSERT INTO Customer (FirstName, LastName, Address1, Address2, City, State, ZIP, Phone1,
Phone2, ReferredBy, CustomerSince, Notes, PrefersMowing, PrefersLandscaping,
PrefersOther)

VALUES

('John', 'Smith', '100 Maple St', "", 'Longmont', 'MO', '63030', '314-111-2222', '314-111-3333',
'Google', '2023-01-10', "", 1, 0, 0),
(('Sara', 'Johnson', '222 Oak Ave', "", 'Longmont', 'MO', '63030', '314-444-5555', '314-666-7777',
'Referral', '2023-03-05', "", 0, 1, 0),
(('Emily', 'Nguyen', '300 Elm Dr', 'Apt 4', 'Longmont', 'MO', '63030', '314-888-9999',
'314-000-1111', "", '2024-02-20', "", 1, 1, 0),
(('Carlos', 'Martinez', '455 Pine Rd', "", 'Longmont', 'MO', '63030', '314-222-3333', '314-444-6666',
'Instagram', '2024-01-10', "", 0, 0, 1);

-- Employees related to sales (working the register)

```
INSERT INTO Employee (FirstName, LastName, Role, Phone)
VALUES
('Rachel', 'Marshall', 'Office Manager', '314-555-0001'),
('Tom', 'Flynn', 'Retail Manager', '314-555-0002');

-- 4 Retail Sales Tickets
INSERT INTO Sale (CustomerID, SaleDate, SoldByEmployeeID, SaleType, DeliveryCharge,
SalesTax, LandscapingRelated, SpecialInstructions)
VALUES
(1, '2024-04-01', 1, 'InStore', 0.00, 2.45, 0, 'Requested smaller bags.'),
(2, '2024-04-02', 2, 'InStore', 0.00, 3.10, 0, 'Assisted by Rachel.'),
(3, '2024-04-03', 1, 'InStore', 0.00, 4.25, 0, 'Picked up in afternoon.'),
(4, '2024-04-04', 2, 'InStore', 0.00, 3.80, 0, 'Used loyalty card.');

-- Sale Items
INSERT INTO SaleItem (SaleID, ItemSKU, Quantity, UnitPrice, EmployeeID)
VALUES
-- Ticket 1
(1, 'TREE001', 1, 55.00, 1),
(1, 'SHRB001', 2, 30.00, 1),
(1, 'MULCH01', 1, 20.00, 1),
(1, 'GRVL01', 1, 30.00, 1),

-- Ticket 2
(2, 'TREE002', 1, 40.00, 2),
(2, 'GRVL01', 2, 32.00, 2),
(2, 'STONE01', 1, 90.00, 2),
(2, 'MULCH02', 1, 21.00, 2),

-- Ticket 3
(3, 'PER001', 3, 20.00, 1),
(3, 'TREE003', 1, 45.00, 2),
(3, 'MULCH01', 2, 22.00, 1),
(3, 'STONE02', 1, 88.00, 2),

-- Ticket 4
(4, 'TREE003', 1, 50.00, 2),
(4, 'PER002', 2, 25.00, 2),
(4, 'GRVL02', 1, 33.00, 2),
(4, 'STONE01', 1, 88.00, 1);
```

Querying against database

Customer Data:

```
SELECT
    CONCAT(FirstName, ' ', LastName) AS FullName,
    Address1,
    Address2,
    City,
    State,
    ZIP,
    Phone1,
    Phone2
FROM Customer;
```

Plant Data:

```
SELECT
    Type AS Category,
    ItemSKU,
    Name AS Description,
    Size,
    QtyAvailable
FROM InventoryItem
WHERE Type IN ('Tree', 'Shrub', 'Perennial');
```

Landscaping Material:

```
SELECT
    Type AS MaterialCategory,
    ItemSKU,
    Name AS Description,
    UnitOfMeasure,
    QtyAvailable
FROM InventoryItem
WHERE Type IN ('Mulch', 'Gravel', 'Topsoil', 'Stone', 'Paver', 'Timber', 'Sand');
```

Seeing what landscaping material belongs to which supplier

```
SELECT
    I.ItemSKU,
```

```

I.Name AS Description,
I.Type AS MaterialCategory,
I.SupplierID,
S.Name AS SupplierName
FROM InventoryItem I
JOIN Supplier S ON I.SupplierID = S.SupplierID
WHERE I.Type IN ('Mulch', 'Gravel', 'Topsoil', 'Stone', 'Paver', 'Timber', 'Sand');

```

Previously Final Project Submitted Proposal (Raw Notes):

1. The database management system being used:
 - I will be using MySQL and will be accessing it via MySQL Workbench
2. Your selection of Case Study to complete.
 - I will be choosing Eden Case Study
3. Describe the scope of your solution and details about the tables you will be creating.
 - The scope of my solution will be to perform the task of a consultant and organize all of the data that Frank has and provide a better view of his company, and provide analysis on where he can improve his business.
 - Database design must support the following business processes at Eden:
 - The ordering of plants and landscaping materials from suppliers
 - Taking physical inventory to see what's in stock
 - Receiving incoming shipments of plants and landscaping materials from suppliers
 - Fulfilling customer orders and purchases, including landscaping jobs
 - In order to do that, my project scope **will** include (The SCOPE of this assignment includes your database design for ONLY the inventory kept by Eden for their **LIVE PLANT STOCK and LANDSCAPING MATERIALS**):
 - **Plants and landscaping materials** that are sold to customers who come to the store, make a purchase, and pick up and **carry out their purchased items (in-store purchase)**
 - **Plants and landscaping materials** that are sold to customers who come to the store, make a purchase, and have Eden **deliver their purchased items to their home (delivery purchase)**
 - **Plants and landscaping materials** that are sold to customers as part of a **landscaping service contract ("job")**, where Eden loads up a truck and sends a crew to the customer's home to do a landscaping project
 - Stuff that customers need that they didn't have already and the contract requires those things
 - Scope of my project will **not** include:

- Tracking of any items kept in inventory for the **purpose of retail sales through Eden's retail store** (like shovels, rakes, seeds, bags of fertilizer, hoses, etc.)
 - Any inventory of tools and **equipment used by Eden's crews** as they go out to customers' locations and **do landscaping jobs** (like the backhoe, wheelbarrows, chain saws, shovels, etc.)
4. Include a document or drawing depicting your database design using the information in the case study as your input. Your design should include
- a. All entities (a person, place, thing or event about which you are keeping data) with proper keys defined
 - b. All attributes, by entity, with data type, length, and constraints defined as appropriate
 - c. If your design is relational, your database design (Data Model or ERD) must show all relationships between tables showing captions (1-way is OK), and proper optionality and cardinality.
 - i. If your design is NOT relational, a drawing will suffice. You should show how the entities in your database are related to each other.

Ben's notes:

- Since this project scope will not include things like inventory for retail sales or inventory of equipment and tools used for landscaping, we should also only care about employees that are performing these tasks and not employees who are for example working the front desk register
- Same with supplier list, some suppliers that are retail-only goods (like hoses, seed bags, etc.) do not need to be included and we do not need to track every tiny item in retail store stock, but we do need to track materials and where they came from relevant to the landscaping materials so things like mulch, gravel, stone, trees, shrub, that are not already in stock
 - And we should keep a record of supplier names, categories, contacts, and orders in addition to things like cost tracking and inventory sourcing
 - We probably don't need insecticides and herbicides since these are not directly under the landscaping job scope (meaning if these are not there, it does not determine whether the job is "finished")
 - We can remove "Security Products Company" because that's not really landscaping, and so products sold for this are not considered a part of landscaping materials
- According to the advertisement, 2 lawn mowing crews + 2 landscaping crews operate within Eden
- Eden services include complete turf maintenance, **landscaping**, tree care, **retaining wall installation**, **irrigation system**, leaf removal, gutter clean, snow removal

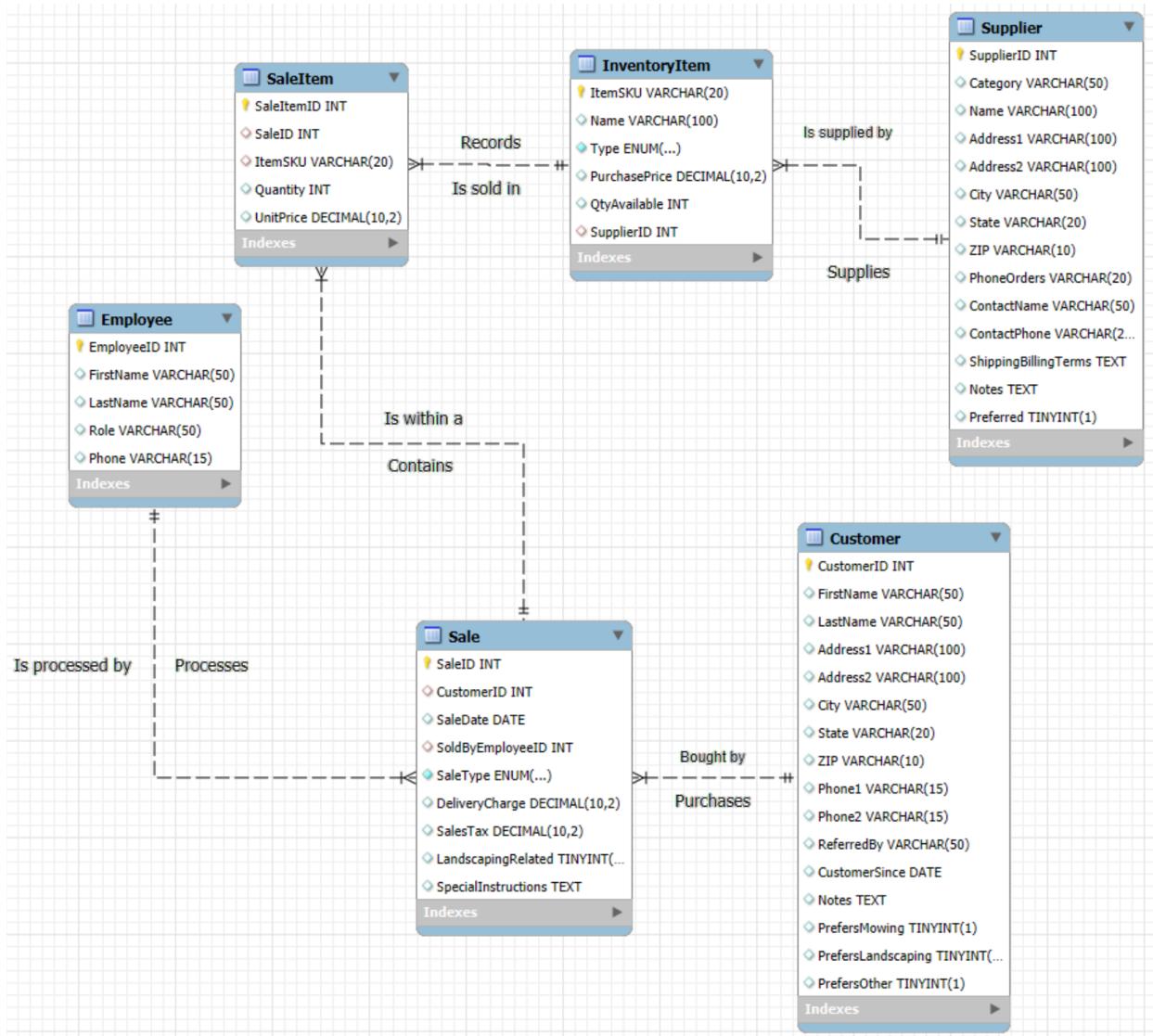
- But we only care about “**Landscaping**”, “**Retaining Wall Installation**” (because necessary for shaping gardens, managing slopes, etc.), “**Irrigation System**” (because might need to reroute plumbing for landscaping)
- Since turf maintenance is granular stuff, I don’t consider that landscaping-related
- We probably don’t need a Bill of Lading because that’s typically for large freight orders. One might argue that Eden might be needing to track backorders and third-party shippers, but I’m choosing to not include it because our database focuses on materials **used** in landscaping jobs and customer purchases. So not for future orders, but things that are already here, so no need for things like backorder because that would be for the future.
 - Second, I don’t see Eden being an international importer of high-end fancy goods that could only be found in certain areas of the world that would require formal Bill of Lading documentation passing through customs. It’s very likely that this mom-and-pop shop is sourcing all their goods domestically. Therefore, all the basic information we need about order time and tracker/supplier orders can be found via the other forms like Supplier Record and Supplier List.
 - I don’t include which employee orders from which supplier. This probably is not necessary since no bill of lading confirmation tracking like the employee process is being modeled.
- I would also argue that the property of Eden’s company (Map of Eden Property) has nothing to do with customer landscaping or sale of plants & landscaping materials, it’s just noise
- “Order Log” also contains information that already exists inside Supplier Record and Supplier List, but the most important thing that Order Log has is that it shows the amount of goods and price it was sold at, and potentially which employee processed it, etc. so this is probably information we don’t want to lose
- We’re not primarily tracking the details of landscaping services themselves, but rather tracking **plants and landscaping materials sold to customers**, indicating whether those **materials were part of a landscaping job** (vs. in-store pickup or delivery), and the specific type of service (e.g., mulching, pruning) is **not central** to the schema
- The employee skills list is a list of employees (other than Frank) and their specialty to landscaping service, but don’t immediately contribute to the sale of plants and landscaping materials, therefore are not central to the schema, which means we can remove them from the core tables
 - They’re good context to have about what kind of landscaping materials *may* be sold to support these landscaping services, but they’re not within the direct scope of this project since the project is just tracking the sales of plants and landscaping materials.
- In my ERD, we don’t need to add pronouns like “he/they” or names like “Rachel”, “Frank”, etc. We don’t need to include bios or skills, we just need to create the skeleton so that if there was an Eden database, that if we test it against that database, our query will pull the right data.
- We are also not tracking specific customer ongoing balances like credit/debits, which is acceptable because the available information in the pictures are 1. Not sufficient enough

to track that 2. Not super core to our project's scope (since our project scope seems to be more on an individual transaction basis rather than an ongoing multi-year + multi-transaction development basis).

- As a consultant to the business, it's important to communicate clear and concise information

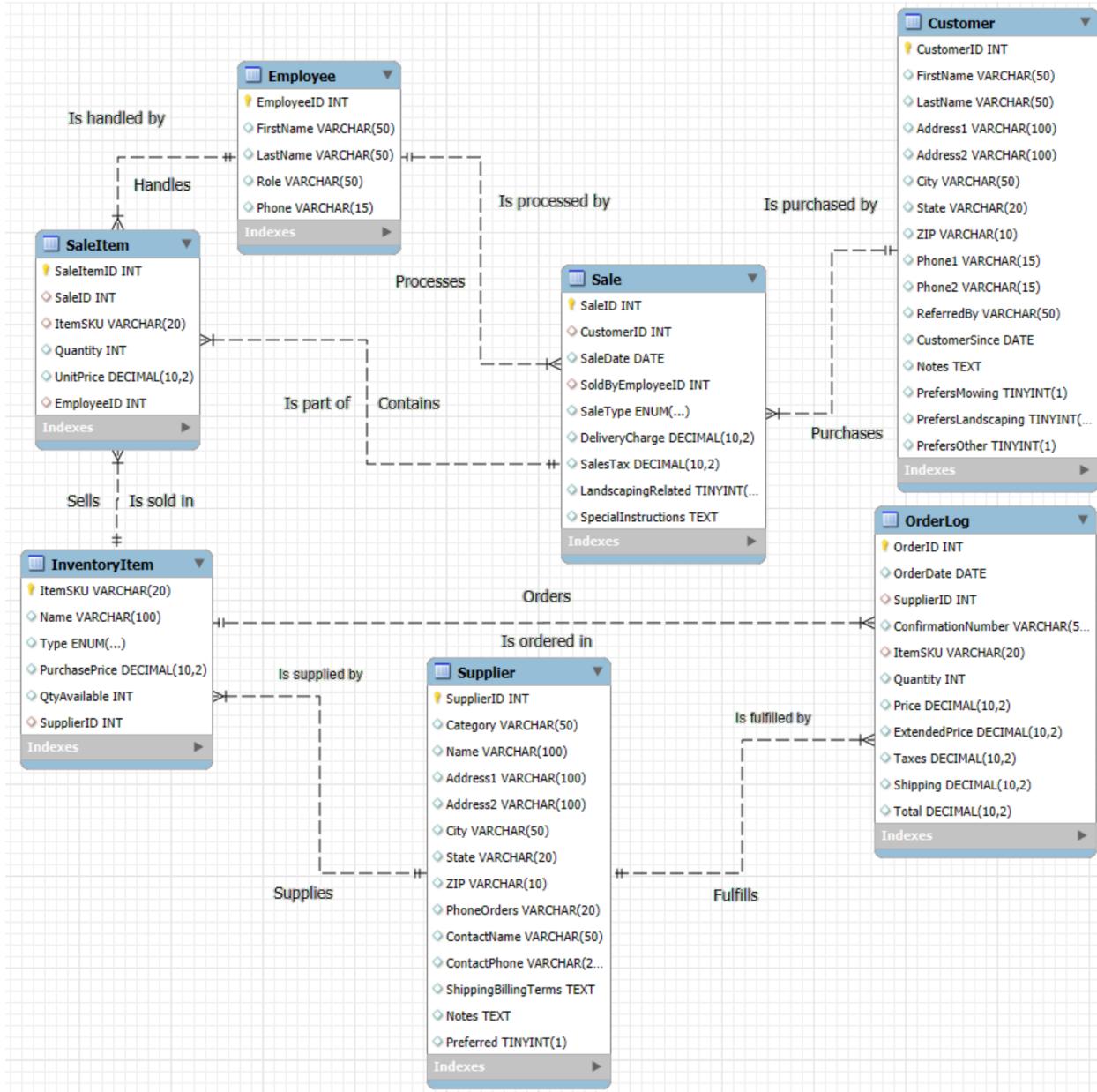
Database Design

Iteration 1 (which was my original idea)



- I felt like it was important to draw the relationship between Order Log and Supplier and how that affects the item inventory so I added it to the database design (in the next iteration)
- I also felt that this iteration wasn't enough to tell the whole picture

Iteration 2 (the ERD I'd like to submit for my proposal)



- Supplier supplies InventoryItem
 - → Each Supplier supplies one or more InventoryItem records.
 - ← Each InventoryItem is supplied by one Supplier.
- Supplier fulfills OrderLog
 - → Each Supplier fulfills one or more OrderLog entries.
 - ← Each OrderLog is fulfilled by one Supplier.

- **InventoryItem** is ordered in **OrderLog**
 - → Each **InventoryItem** is ordered in one or more **OrderLog** entries.
 - ← Each **OrderLog** orders exactly one **InventoryItem**.
- **InventoryItem** is sold in **SaleItem**
 - → Each **InventoryItem** is sold in one or more **SaleItem** entries.
 - ← Each **SaleItem** sells one **InventoryItem**.
- **Sale** contains **SaleItem**
 - → Each **Sale** contains one or more **SaleItem** entries.
 - ← Each **SaleItem** is part of one **Sale**.
- **Customer** purchases **Sale**
 - → Each **Customer** purchases one or more **Sale** entries.
 - ← Each **Sale** is purchased by one **Customer**.
- **Employee** processes **Sale**
 - → Each **Employee** processes one or more **Sale** records.
 - ← Each **Sale** is processed by one **Employee**.
- **Employee** handles **SaleItem**
 - → Each **Employee** handles one or more **SaleItem** entries.
 - ← Each **SaleItem** is handled by one **Employee**.