Flour & Flowers✿

Contribution % of each team member

Names	Contribution %
Soundarya Baskar	**25**
Varsha George	**25**
Anchal Sachdev	**25**
Palak Bhargava	**25**

Problem Domain and Solution

Miniworld: Flour&Flowers

There is one business Flour&Flowers. The scope of this example includes customers, suppliers, and facilities. Out of scope are external delivery service and partner sellers.

Business Rules

1. Each customer must have a name, a phone number, an address, email, payment method. A customer may place several orders to be fulfilled.

2. The orders to be fulfilled must be fulfilled within a week.

3. Each supplier (flowers or ingredients) has a representative, which has a name, phone number, email, and address of business

4. Each employee will have a name (first, middle, last), employee ID, manager ID, email, phone number, job type.

5. Each manager will have a name (first, middle, last), employee ID, phone number, email, phone number, job type.

6. Each employee will be either an hourly employee or a salaried employee based on job type.

7. The storage inventory consists of what supplies we have, how much of these supplies we have, and what type of products (baked goods or flowers) they are used for.

8. Each store will offer a variety of products, like Pops&Pansies, Cupcakes&Chrysanthemums, RedVelvet&Roses, etc.

9. The business purchases several types of flowers from the suppliers. These will include flowers such as Roses, Pansies, etc. Each of these flowers will have attributes such as colors, price, type, etc.

Major Entity Types

- **Flowers**

  flowers are one of the products our business will sell, which will include a variety of flowers. It is a core component of our mini-world, and it will be supplied with every order purchased. Flowers interact with the baked goods as they will be sold with the baked goods, purchased from the suppliers, and be stored in the facilities.

- **Baked goods**

  The baked goods entity will include the different baked goods that will be sold. The baked goods are a core component of the mini-world and will be included in every purchase. The baked goods entity type interacts with the Flowers entity type as they will be sold together, purchased from the suppliers, and be stored in the facilities.

- **Suppliers**

  The suppliers will include the providers of raw ingredients. These include flour, sugar, eggs, baking powder suppliers, etc. The suppliers also include the suppliers of flowers and other bouquet-making materials. The Suppliers will interact with the Manager as the manager deals with inventory and stock.

- **Customers**

  A customer is a consumer that buys goods from our facility. We will collect the customer’s name, contact information, and payment method to fulfill the order that they placed.

- **Orders**

  The customer will place an order on our website or in-person at our facility, and we will be in charge of providing these customers with the products that they desire to buy. We will link an order id to a list of products that were ordered.

- **Employees**

  The employee is someone who works at the facility for a wage, and will have several job types. They will help bake the goods, put the product together, and serve them to our customers. The employees also have a facility that they work in.

Products Offered

The products offered are what we sell to our customers. These are the premade baked goods and flower combos that are available for customers to purchase.

Storage Inventory

The storage inventory contains information on what supplies we have and how much of the supplies are available to use. It includes flowers and supplies for baked goods.

Assumptions on how many tables, relationships discovered, and volumes of data

How many tables

We assume we will have around 8 tables.

- The tables consist of CUSTOMER, ORDER, SUPPLIER, EMPLOYEE, STORAGE_INVENTORY, BAKED_GOODS, FLOWERS, PRODUCTS_OFFERED.

How many relationships

We assume there will be 8 relationships.

The relationships will consist of

- A Customer places an Order, a Baked_good is a Product, Flowers are a Product, an Order can consist of one or more Products, Employees handle the order, Employees that are managers manage the supplier, A Supplier provides supplies for the storage inventory, A Product is made using supplies in the storage inventory.

Volumes of data

- We estimate that we will have 500 orders per year and employ 10 employees a year.

Summary

- This business will sell flowers with baked goods. We will provide information on orders, customers, inventory, product offerings, and employees to the users of our database.

Data Dictionary for the Schema

<table>
<tr><th>Table</th><th>Table Description</th><th>Primary Key</th><th>SuperKey</th><th>Attributes</th><th>Foreign Key</th></tr>
<tr><td>Customer</td><td>A customer is a consumer that buys goods from our facility.</td><td>Customer_ID</td><td>(Customer_ID, Name, C_Phone, C_Email, Payment, ~~Order_ID~~)</td><td>
<table>
<tr><td>Attribute</td><td>Data Type</td><td>Domain</td></tr>
<tr><td>Customer_ ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
<tr><td>Name</td><td>VARCHAR</td><td>Any string</td></tr>
<tr><td>C_Phone_ no</td><td>BIGINT, width 10</td><td>Any positive integer</td></tr>
<tr><td>C_Email</td><td>VARCHAR</td><td>Any string</td></tr>
</table>
</td><td>~~Order_ID~~</td></tr>
<tr><td>Orders</td><td>The goods the customer buys from our facility.</td><td>Order_ID</td><td>(Order_ID, Flowers, Baked_goods, Date, Customer_ID)</td><td>
<table>
<tr><td>Attribute</td><td>Data Type</td><td>Domain</td></tr>
<tr><td>Order_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
<tr><td>Flowers</td><td>VARCHAR</td><td>Any string</td></tr>
<tr><td>Baked_ goods</td><td>VARCHAR</td><td>Any string</td></tr>
<tr><td>Date</td><td>DATETIME</td><td>Current date</td></tr>
<tr><td>Customer_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
</table>
</td><td>Customer_ID</td></tr>
<tr><td>Supplier</td><td>The providers of the raw ingredients to make the goods.</td><td>Supplier_ID</td><td>(Supplier_ID, Supply_Type, Supplier_Name, S_Email, S_Phone~~_#~~)</td><td>
<table>
<tr><td>Attribute</td><td>Data Type</td><td>Domain</td></tr>
<tr><td>Supplier _ID</td><td>TINYINT, width 3</td><td>Integers greater</td></tr>
</table>
</td><td></td></tr>
</table>

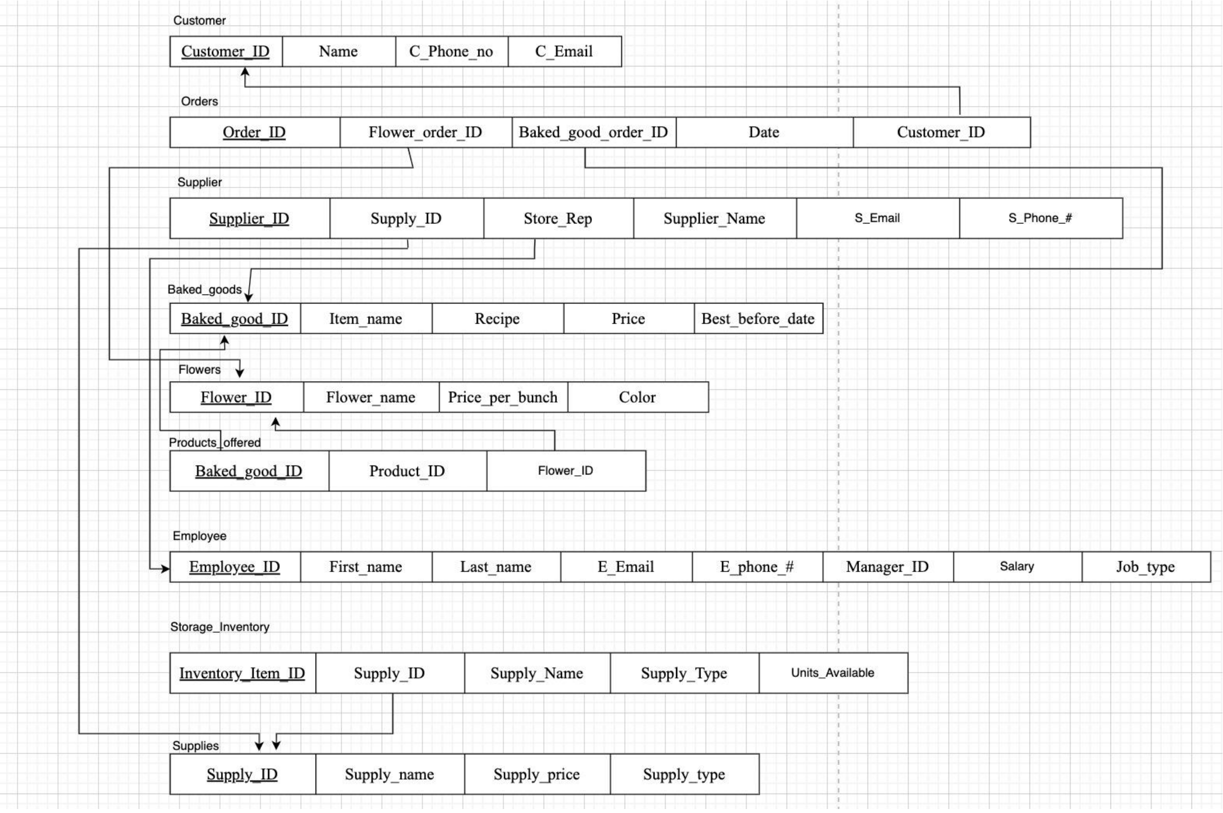
<table>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>than 0</td></tr>
<tr><td>Supply_ID</td><td>SET</td><td>Several values from a different list of supplies</td></tr>
<tr><td>Store_Rep</td><td>TINYINT(3)</td><td>String up to 3 letters</td></tr>
<tr><td>Supplier_Name</td><td>VARCHAR</td><td>Any String</td></tr>
<tr><td>S_Phone~~_#~~</td><td>BIGINT</td><td>Width 10</td></tr>
<tr><td>S_Email</td><td>VARCHAR</td><td>Any String</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Baked_goods</td>
<td>The baked goods such as cookies, cupcakes, cakes, etc. that the store will sell</td>
<td>Baked_good_ID</td>
<td>(Baked_good_ID, Item_name, Recipe, Price, Best_before_date)</td>
<td>
<table>
<tr><td>Attributes</td><td>Data Type</td><td>Domain</td></tr>
<tr><td>Baked_good_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
<tr><td>Item_name</td><td>VARCHAR</td><td>Any string, unique</td></tr>
<tr><td>~~Supplies~~</td><td>~~SET~~</td><td>~~Several values from given list~~</td></tr>
<tr><td>Recipe</td><td>VARCHAR</td><td>Any string</td></tr>
<tr><td>Price</td><td>FLOAT</td><td>size, 4</td></tr>
<tr><td>Best_before_date</td><td>DATE</td><td></td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Flowers</td>
<td>The different flowers that the store has available to sell. These</td>
<td>Flower_ID</td>
<td>(Flower_ID, Flower_name, Price_per_bunch, Color)</td>
<td>
<table>
<tr><td>Attribute</td><td>Data Type</td><td>Domain</td></tr>
<tr><td>Flower_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
</table>
</td>
<td></td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>flowers can be used to create bouquets that go with the final product.</td>
<td></td>
<td></td>
<td><table>
<tr><td>Flower_name</td><td>ENUM</td><td>Any value from the enum</td></tr>
<tr><td>Price_per_bunch</td><td>FLOAT</td><td>size, 2</td></tr>
<tr><td>Color</td><td>ENUM</td><td>Any value from the enum</td></tr>
</table></td>
<td></td>
</tr>
<tr>
<td>Products_offered</td>
<td>These are the premade baked goods and flower combos that are available for customers to purchase.</td>
<td>Product_ID</td>
<td>(Product_ID, Baked_good_ID, Flower_ID)</td>
<td><table>
<tr><th>Attribute</th><th>Data Type</th><th>Domain</th></tr>
<tr><td>Baked_good_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
<tr><td>Product_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
<tr><td>Flower_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
</table></td>
<td>Baked_good_ID, Flower_ID</td>
</tr>
<tr>
<td>Employee</td>
<td>The employees work at the facility for a wage, and there are several job types. This table contains information on all the employees of the facility</td>
<td>Employee_ID</td>
<td>(Employee_ID, First_name, Last_name, E_Phone, E_Email, Job_Title, Manager_ID, Salary, Job_type)</td>
<td><table>
<tr><th>Attribute</th><th>Data Type</th><th>Domain</th></tr>
<tr><td>Employee_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
<tr><td>First_name</td><td>VARCHAR (50)</td><td>Any String</td></tr>
<tr><td>Last_ name</td><td>VARCHAR (50)</td><td>Any String</td></tr>
<tr><td>E_Email</td><td>VARCHAR (15)</td><td>Any String</td></tr>
<tr><td>E_Phone</td><td>BIGINT, width 10</td><td>Any positive integer</td></tr>
<tr><td>Manager_ID</td><td>INT, width 5</td><td>Integers greater than</td></tr>
</table></td>
<td>Employee_ID</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td></td><td>0</td></tr>
<tr><td>Salary</td><td>INT, width 6</td><td>Integers greater than 0</td></tr>
<tr><td>Job_Type</td><td>ENUM</td><td>Any value from the enum</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Storage_Inventory</td>
<td>This contains information on what supplies and how much of the supplies are available to use.</td>
<td>Supply_ID</td>
<td>(Inventory_Item_ID, Supply_id, Supply_Name, units_available, Supply_type)</td>
<td>
<table>
<tr><td>Attribute</td><td>Data Type</td><td>Domain</td></tr>
<tr><td>Inventory_Item_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
<tr><td>Supply_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
<tr><td>Supply_Name</td><td>VARCHAR</td><td>Any string</td></tr>
<tr><td>Supply_Type</td><td>ENUM</td><td>Any value from enum</td></tr>
<tr><td>Units_Available</td><td>INT</td><td>Integers greater than 0</td></tr>
</table>
</td>
<td>Supply_ID</td>
</tr>
<tr>
<td>Supplies</td>
<td>This contains information on what supplies and how much of the supplies are available to use.</td>
<td>Supply_ID</td>
<td>(Supply_id, Supply_Name, Supply_price, Supplier_type)</td>
<td>
<table>
<tr><td>Attribute</td><td>Data Type</td><td>Domain</td></tr>
<tr><td>Supply_ID</td><td>TINYINT, width 3</td><td>Integers greater than 0</td></tr>
<tr><td>Supply_Name</td><td>VARCHAR</td><td>Any string</td></tr>
<tr><td>Supply_price</td><td>FLOAT(4,2)</td><td>Any value from enum</td></tr>
<tr><td>Supply_type</td><td>SET</td><td>Integers greater than 0</td></tr>
</table>
</td>
<td>Supply_ID</td>
</tr>
</table>

Supporting Schema Diagram

Tools Used: draw.io



Customer

Customer_ID	Name	C_Phone_no	C_Email

Orders

Order_ID	Flower_order_ID	Baked_good_order_ID	Date	Customer_ID

Supplier

Supplier_ID	Supply_ID	Store_Rep	Supplier_Name	S_Email	S_Phone_#

Baked_goods

Baked_good_ID	Item_name	Recipe	Price	Best_before_date

Flowers

Flower_ID	Flower_name	Price_per_bunch	Color

Products_offered

Baked_good_ID	Product_ID	Flower_ID

Employee

Employee_ID	First_name	Last_name	E_Email	E_phone_#	Manager_ID	Salary	Job_type

Storage_Inventory

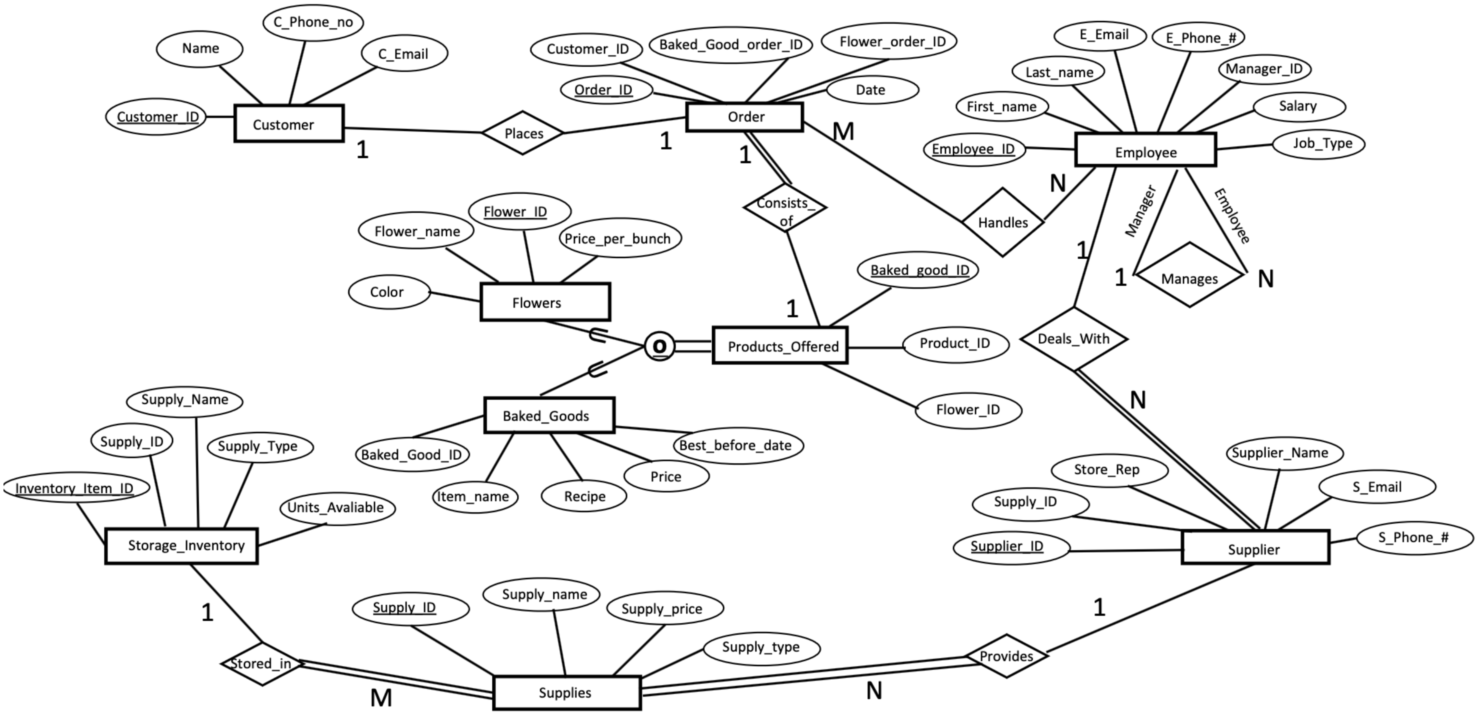
Inventory_Item_ID	Supply_ID	Supply_Name	Supply_Type	Units_Available

Supplies

Supply_ID	Supply_name	Supply_price	Supply_type

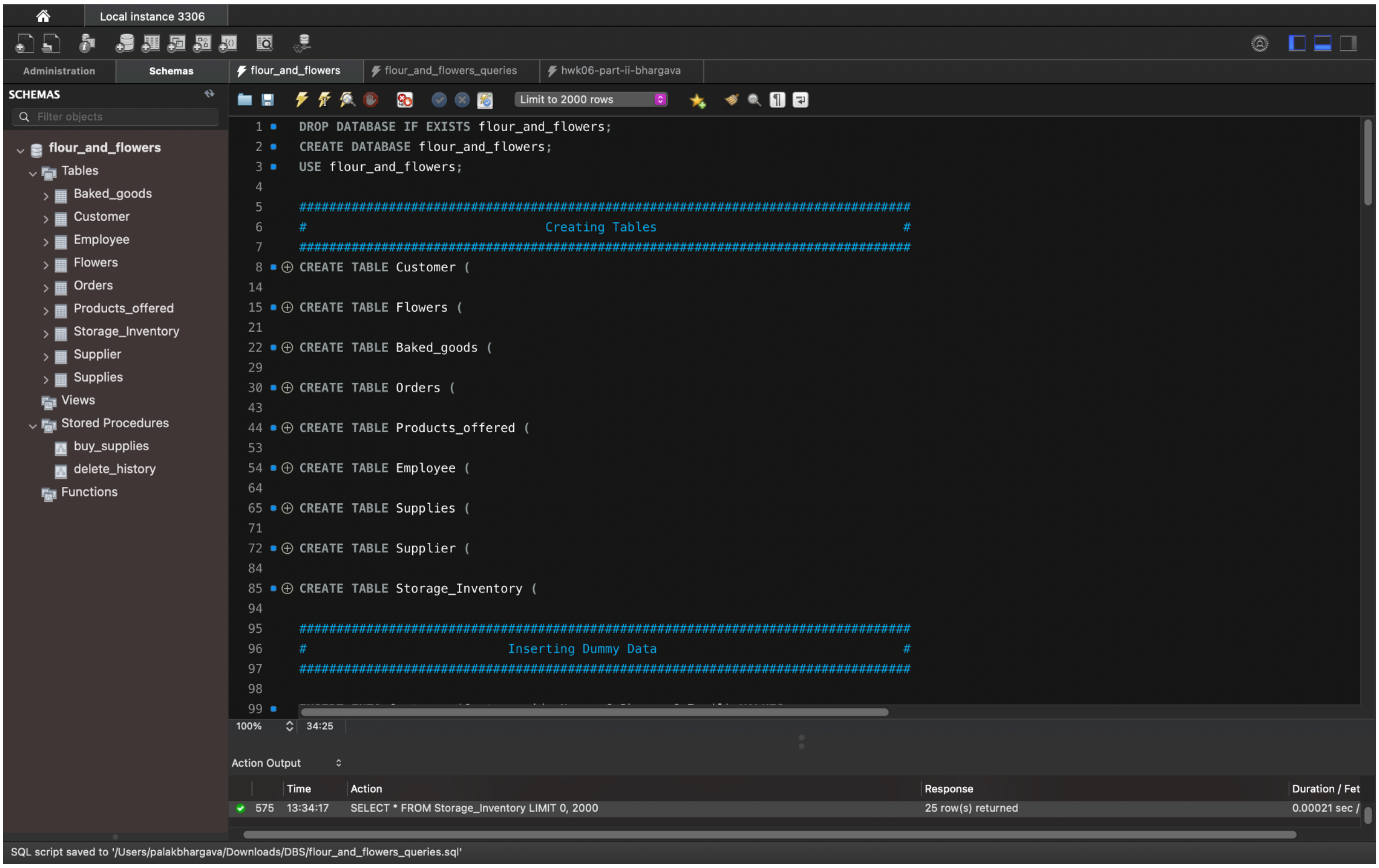
Entity Relationship Diagram (ERD)

Tool Used: MS Powerpoint



Database and SQL

Screenshot of the Database tool work area



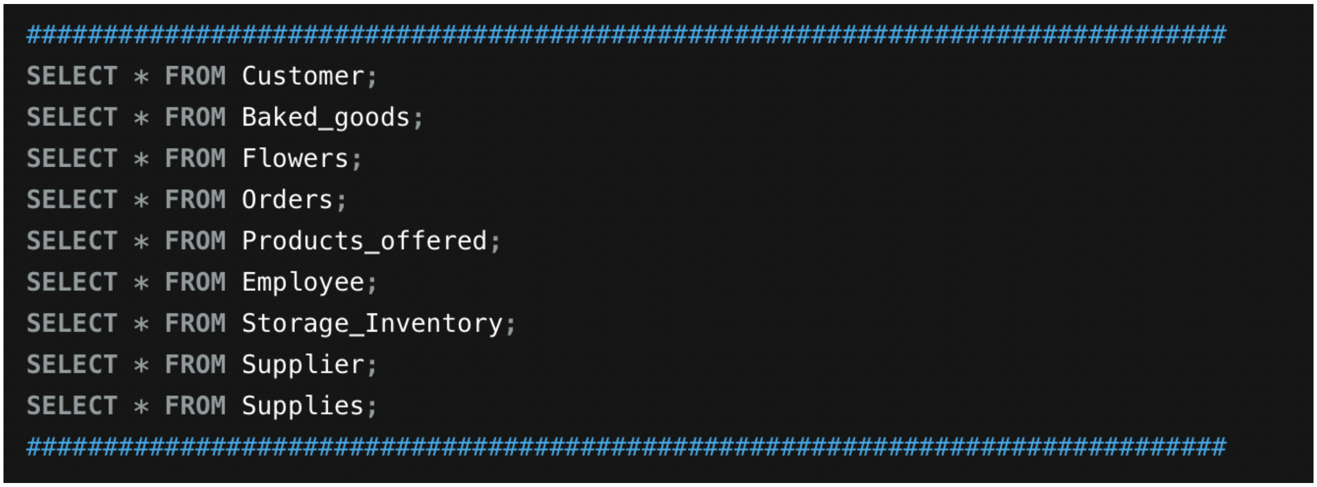
```
DROP DATABASE IF EXISTS flour_and_flowers;
CREATE DATABASE flour_and_flowers;
USE flour_and_flowers;

##################################################################################
#                                Creating Tables                                 #
##################################################################################
CREATE TABLE Customer (

CREATE TABLE Flowers (

CREATE TABLE Baked_goods (

CREATE TABLE Orders (

CREATE TABLE Products_offered (

CREATE TABLE Employee (

CREATE TABLE Supplies (

CREATE TABLE Supplier (

CREATE TABLE Storage_Inventory (

##################################################################################
#                            Inserting Dummy Data                                #
##################################################################################

```

	Time	Action	Response	Duration / Fet
✓ 575	13:34:17	SELECT * FROM Storage_Inventory LIMIT 0, 2000	25 row(s) returned	0.00021 sec /

SQL script saved to '/Users/palakbhargava/Downloads/DBS/flour_and_flowers_queries.sql'

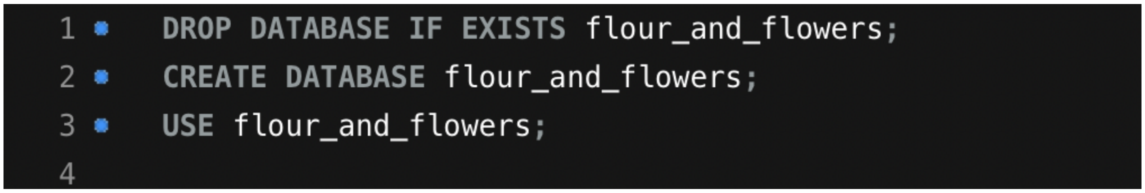
Specific screenshots of how the tool supports SQL

SELECTS:

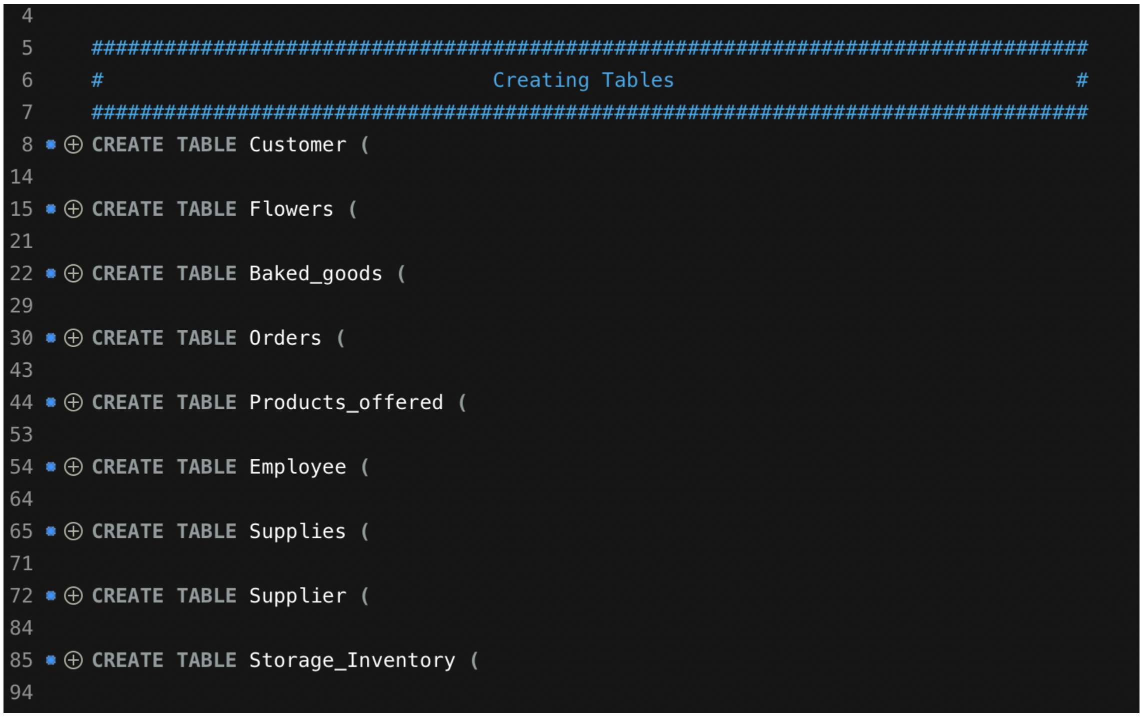


```
##################################################################################
SELECT * FROM Customer;
SELECT * FROM Baked_goods;
SELECT * FROM Flowers;
SELECT * FROM Orders;
SELECT * FROM Products_offered;
SELECT * FROM Employee;
SELECT * FROM Storage_Inventory;
SELECT * FROM Supplier;
SELECT * FROM Supplies;
##################################################################################
```

CREATE, DROP, USE:

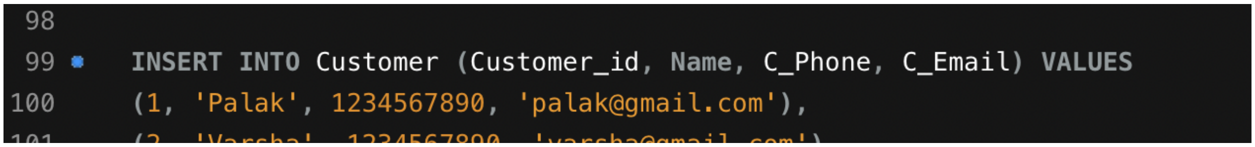


```sql
DROP DATABASE IF EXISTS flour_and_flowers;
CREATE DATABASE flour_and_flowers;
USE flour_and_flowers;

```

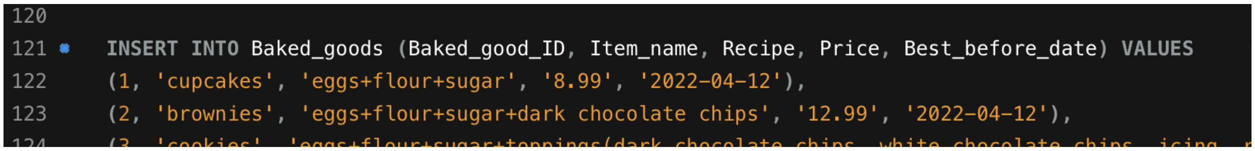


```sql

################################################################################
#                               Creating Tables                                #
################################################################################
CREATE TABLE Customer (

CREATE TABLE Flowers (

CREATE TABLE Baked_goods (

CREATE TABLE Orders (

CREATE TABLE Products_offered (

CREATE TABLE Employee (

CREATE TABLE Supplies (

CREATE TABLE Supplier (

CREATE TABLE Storage_Inventory (

```

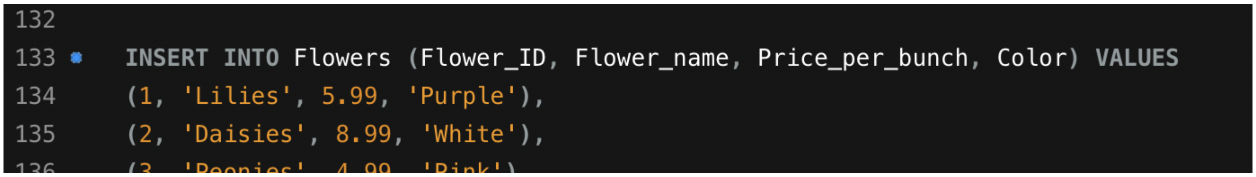
INSERTS:



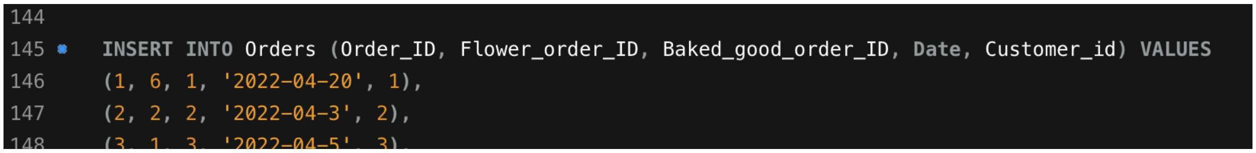
```sql

INSERT INTO Customer (Customer_id, Name, C_Phone, C_Email) VALUES
(1, 'Palak', 1234567890, 'palak@gmail.com'),
(2, 'Varsha', 1234567890, 'varsha@gmail.com'),
```



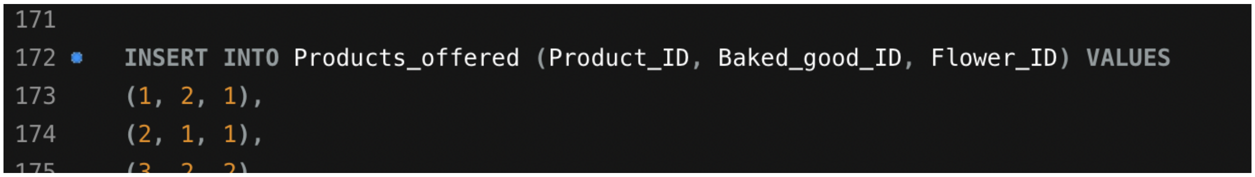
```sql

INSERT INTO Baked_goods (Baked_good_ID, Item_name, Recipe, Price, Best_before_date) VALUES
(1, 'cupcakes', 'eggs+flour+sugar', '8.99', '2022-04-12'),
(2, 'brownies', 'eggs+flour+sugar+dark chocolate chips', '12.99', '2022-04-12'),
(3, 'cookies', 'eggs+flour+sugar+toppings(dark chocolate chips, white chocolate chips, icing, r
```



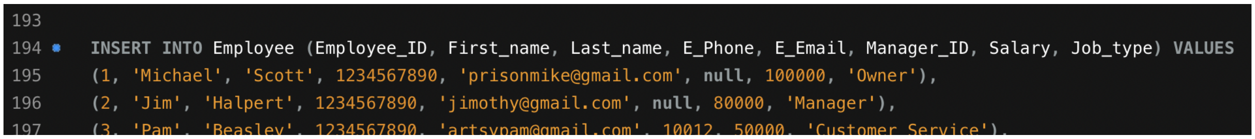
```sql

INSERT INTO Flowers (Flower_ID, Flower_name, Price_per_bunch, Color) VALUES
(1, 'Lilies', 5.99, 'Purple'),
(2, 'Daisies', 8.99, 'White'),
(3, 'Peonies', 4.99, 'Pink'),
```



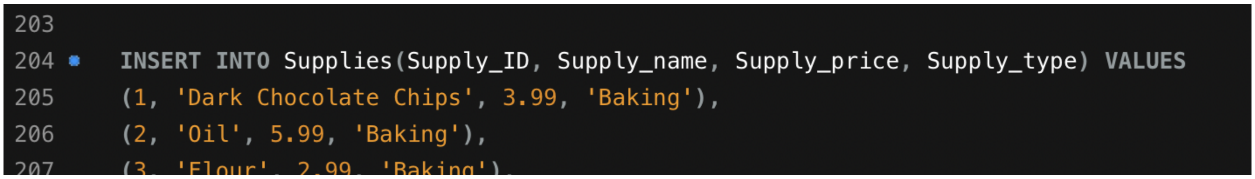
```sql

INSERT INTO Orders (Order_ID, Flower_order_ID, Baked_good_order_ID, Date, Customer_id) VALUES
(1, 6, 1, '2022-04-20', 1),
(2, 2, 2, '2022-04-3', 2),
(3, 1, 3, '2022-04-5', 3),
```



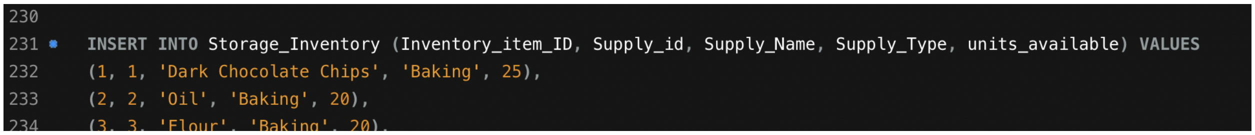
```
INSERT INTO Products_offered (Product_ID, Baked_good_ID, Flower_ID) VALUES
(1, 2, 1),
(2, 1, 1),
(3, 2, 2),
```



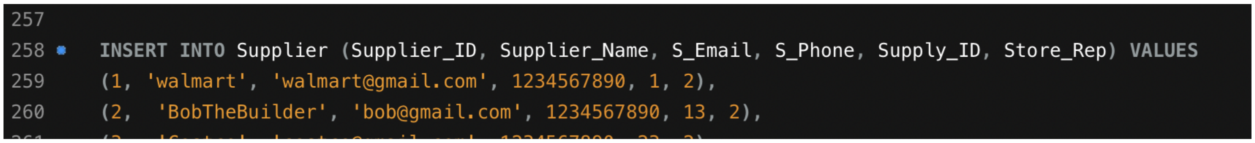
```
INSERT INTO Employee (Employee_ID, First_name, Last_name, E_Phone, E_Email, Manager_ID, Salary, Job_type) VALUES
(1, 'Michael', 'Scott', 1234567890, 'prisonmike@gmail.com', null, 100000, 'Owner'),
(2, 'Jim', 'Halpert', 1234567890, 'jimothy@gmail.com', null, 80000, 'Manager'),
(3, 'Pam', 'Beasley', 1234567890, 'artsypam@gmail.com', 10012, 50000, 'Customer Service'),
```



```
INSERT INTO Supplies(Supply_ID, Supply_name, Supply_price, Supply_type) VALUES
(1, 'Dark Chocolate Chips', 3.99, 'Baking'),
(2, 'Oil', 5.99, 'Baking'),
(3, 'Flour', 2.99, 'Baking'),
```

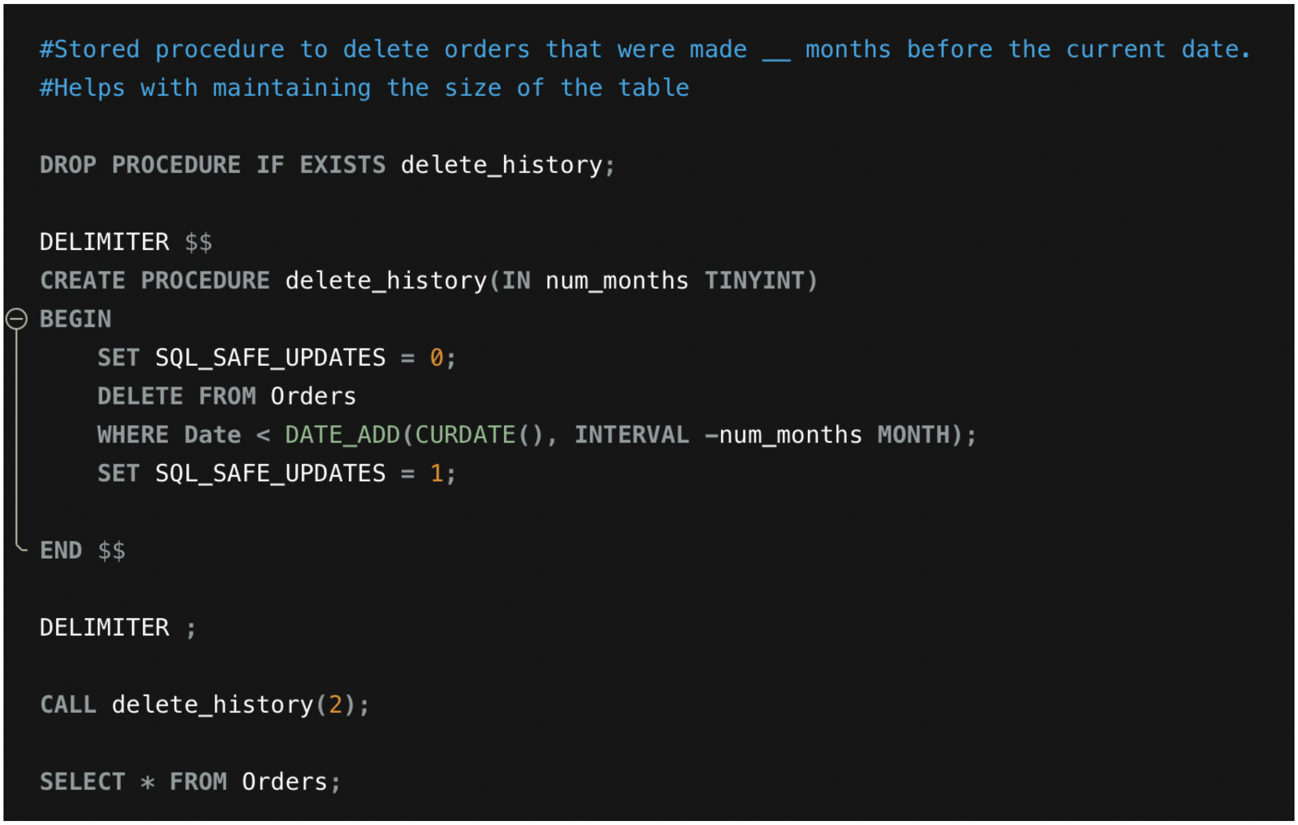


```
INSERT INTO Storage_Inventory (Inventory_item_ID, Supply_id, Supply_Name, Supply_Type, units_available) VALUES
(1, 1, 'Dark Chocolate Chips', 'Baking', 25),
(2, 2, 'Oil', 'Baking', 20),
(3, 3, 'Flour', 'Baking', 20),
```



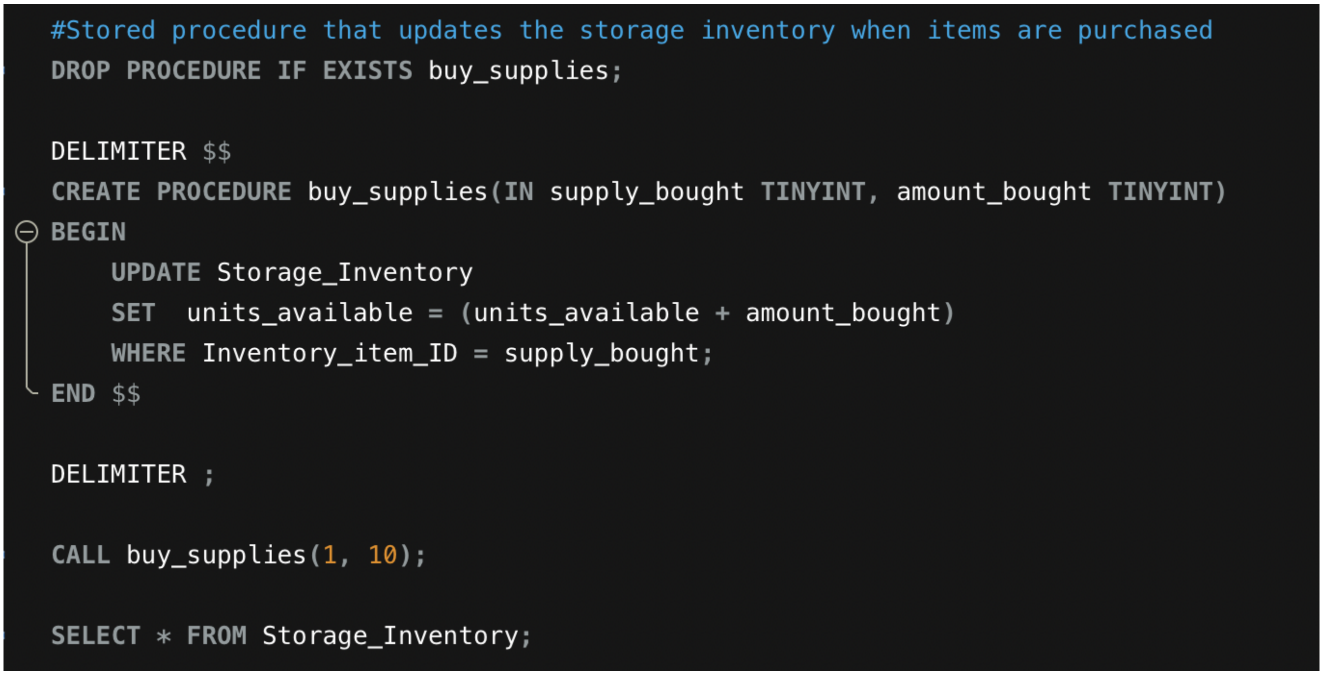
```
INSERT INTO Supplier (Supplier_ID, Supplier_Name, S_Email, S_Phone, Supply_ID, Store_Rep) VALUES
(1, 'walmart', 'walmart@gmail.com', 1234567890, 1, 2),
(2,  'BobTheBuilder', 'bob@gmail.com', 1234567890, 13, 2),
(3,  'Costco', 'costco@gmail.com', 1234567890, 23, 2),
```

DELETE:



```
#Stored procedure to delete orders that were made __ months before the current date.
#Helps with maintaining the size of the table

DROP PROCEDURE IF EXISTS delete_history;

DELIMITER $$
CREATE PROCEDURE delete_history(IN num_months TINYINT)
BEGIN
    SET SQL_SAFE_UPDATES = 0;
    DELETE FROM Orders
    WHERE Date < DATE_ADD(CURDATE(), INTERVAL -num_months MONTH);
    SET SQL_SAFE_UPDATES = 1;

END $$

DELIMITER ;

CALL delete_history(2);

SELECT * FROM Orders;
```

UPDATE:

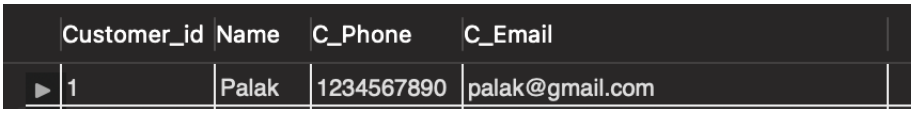
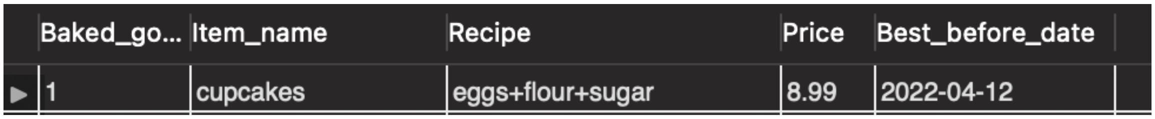
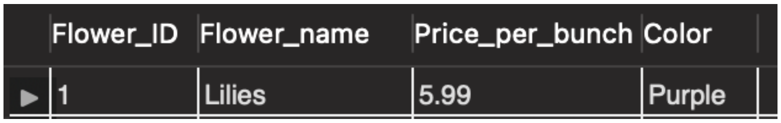
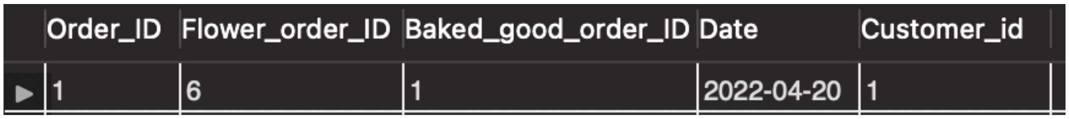
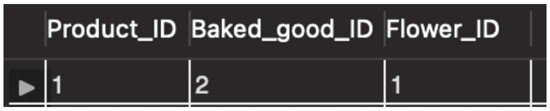
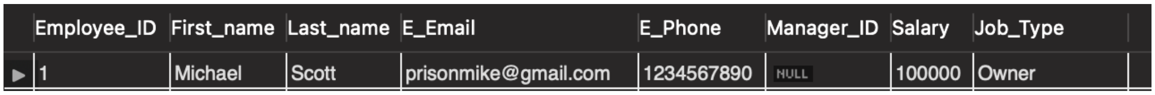
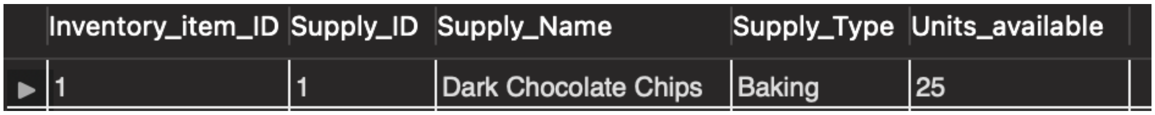
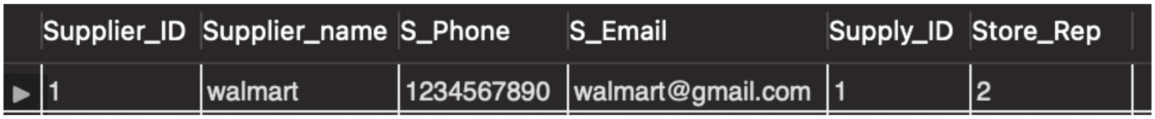
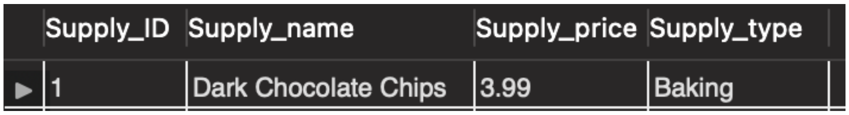


```
#Stored procedure that updates the storage inventory when items are purchased
DROP PROCEDURE IF EXISTS buy_supplies;

DELIMITER $$
CREATE PROCEDURE buy_supplies(IN supply_bought TINYINT, amount_bought TINYINT)
BEGIN
    UPDATE Storage_Inventory
    SET  units_available = (units_available + amount_bought)
    WHERE Inventory_item_ID = supply_bought;
END $$

DELIMITER ;

CALL buy_supplies(1, 10);

SELECT * FROM Storage_Inventory;
```

Data Generation

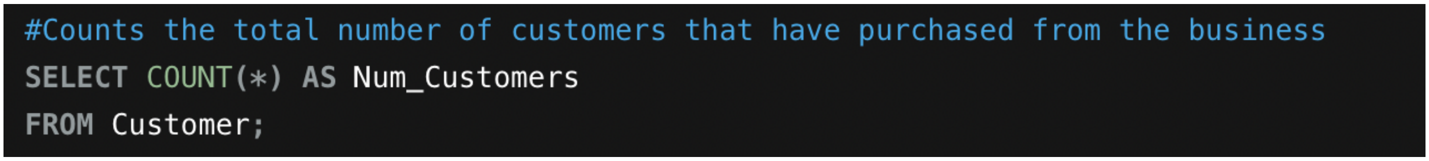
Table Name	**Planned Size**
Customer	3000 customers per year
Baked_goods	10 baked goods (with variations)
Flowers	10 different flowers
Orders	5000 orders per year
Products_offered	120 products (baked good + flower combos, plus flowers and baked good offered individually)
Employee	20 employees
Storage_inventory	40 individual items ranging from 10-50 units each
Supplier	5-10 suppliers
Supplies	40 items

Table Data

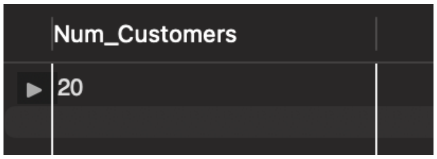
<table>
<tr><td>Table Name</td><td>Row/Tuple with Data</td></tr>
<tr><td>Customer</td><td>
<table>
<tr><th></th><th>Customer_id</th><th>Name</th><th>C_Phone</th><th>C_Email</th></tr>
<tr><td>▶</td><td>1</td><td>Palak</td><td>1234567890</td><td>palak@gmail.com</td></tr>
</table></td></tr>
<tr><td>Baked_goods</td><td>
<table>
<tr><th></th><th>Baked_go...</th><th>Item_name</th><th>Recipe</th><th>Price</th><th>Best_before_date</th></tr>
<tr><td>▶</td><td>1</td><td>cupcakes</td><td>eggs+flour+sugar</td><td>8.99</td><td>2022-04-12</td></tr>
</table></td></tr>
<tr><td>Flowers</td><td>
<table>
<tr><th></th><th>Flower_ID</th><th>Flower_name</th><th>Price_per_bunch</th><th>Color</th></tr>
<tr><td>▶</td><td>1</td><td>Lilies</td><td>5.99</td><td>Purple</td></tr>
</table></td></tr>
<tr><td>Orders</td><td>
<table>
<tr><th></th><th>Order_ID</th><th>Flower_order_ID</th><th>Baked_good_order_ID</th><th>Date</th><th>Customer_id</th></tr>
<tr><td>▶</td><td>1</td><td>6</td><td>1</td><td>2022-04-20</td><td>1</td></tr>
</table></td></tr>
<tr><td>Products_offered</td><td>
<table>
<tr><th></th><th>Product_ID</th><th>Baked_good_ID</th><th>Flower_ID</th></tr>
<tr><td>▶</td><td>1</td><td>2</td><td>1</td></tr>
</table></td></tr>
<tr><td>Employee</td><td>
<table>
<tr><th></th><th>Employee_ID</th><th>First_name</th><th>Last_name</th><th>E_Email</th><th>E_Phone</th><th>Manager_ID</th><th>Salary</th><th>Job_Type</th></tr>
<tr><td>▶</td><td>1</td><td>Michael</td><td>Scott</td><td>prisonmike@gmail.com</td><td>1234567890</td><td>NULL</td><td>100000</td><td>Owner</td></tr>
</table></td></tr>
<tr><td>Storage_inventory</td><td>
<table>
<tr><th></th><th>Inventory_item_ID</th><th>Supply_ID</th><th>Supply_Name</th><th>Supply_Type</th><th>Units_available</th></tr>
<tr><td>▶</td><td>1</td><td>1</td><td>Dark Chocolate Chips</td><td>Baking</td><td>25</td></tr>
</table></td></tr>
<tr><td>Supplier</td><td>
<table>
<tr><th></th><th>Supplier_ID</th><th>Supplier_name</th><th>S_Phone</th><th>S_Email</th><th>Supply_ID</th><th>Store_Rep</th></tr>
<tr><td>▶</td><td>1</td><td>walmart</td><td>1234567890</td><td>walmart@gmail.com</td><td>1</td><td>2</td></tr>
</table></td></tr>
<tr><td>Supplies</td><td>
<table>
<tr><th></th><th>Supply_ID</th><th>Supply_name</th><th>Supply_price</th><th>Supply_type</th></tr>
<tr><td>▶</td><td>1</td><td>Dark Chocolate Chips</td><td>3.99</td><td>Baking</td></tr>
</table></td></tr>
</table>

Data Queries

#1: Show a count of the largest population

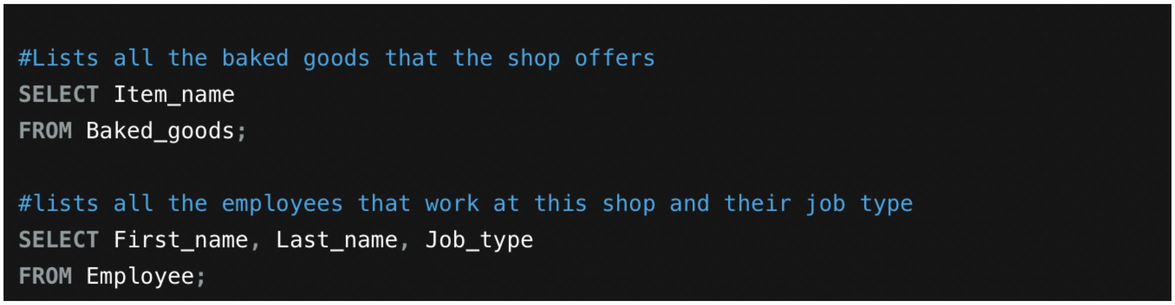


```
#Counts the total number of customers that have purchased from the business
SELECT COUNT(*) AS Num_Customers
FROM Customer;
```



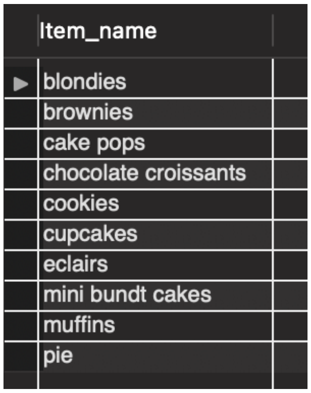
Num_Customers
20

#2: Show a listing of a key entity in the database



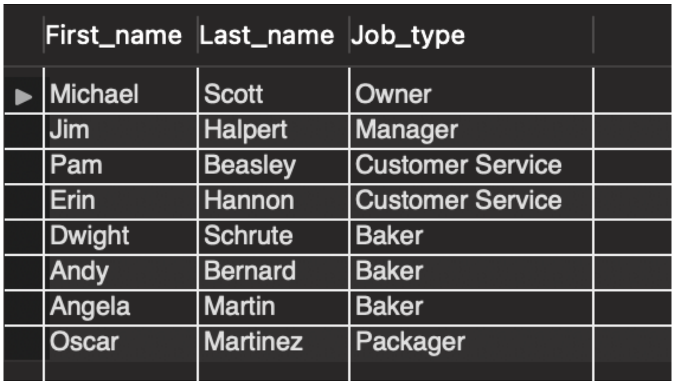
```
#Lists all the baked goods that the shop offers
SELECT Item_name
FROM Baked_goods;

#lists all the employees that work at this shop and their job type
SELECT First_name, Last_name, Job_type
FROM Employee;
```

1.



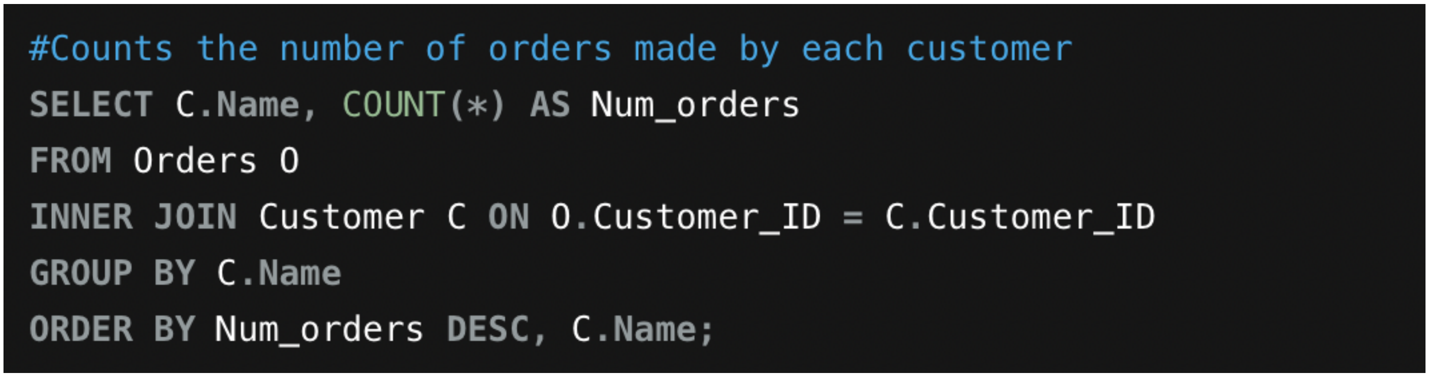
Item_name
blondies
brownies
cake pops
chocolate croissants
cookies
cupcakes
eclairs
mini bundt cakes
muffins
pie

2.

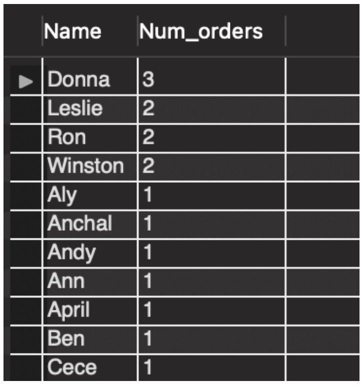


First_name	Last_name	Job_type
Michael	Scott	Owner
Jim	Halpert	Manager
Pam	Beasley	Customer Service
Erin	Hannon	Customer Service
Dwight	Schrute	Baker
Andy	Bernard	Baker
Angela	Martin	Baker
Oscar	Martinez	Packager

#3: Show a list of entities that must function together (a join)

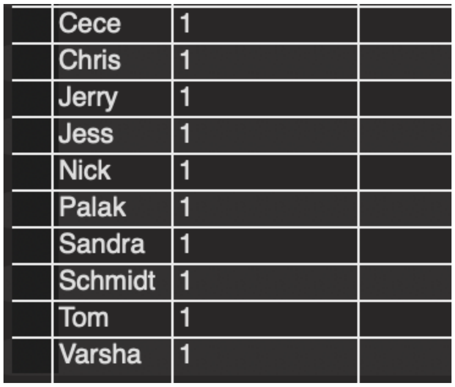


```
#Counts the number of orders made by each customer
SELECT C.Name, COUNT(*) AS Num_orders
FROM Orders O
INNER JOIN Customer C ON O.Customer_ID = C.Customer_ID
GROUP BY C.Name
ORDER BY Num_orders DESC, C.Name;
```



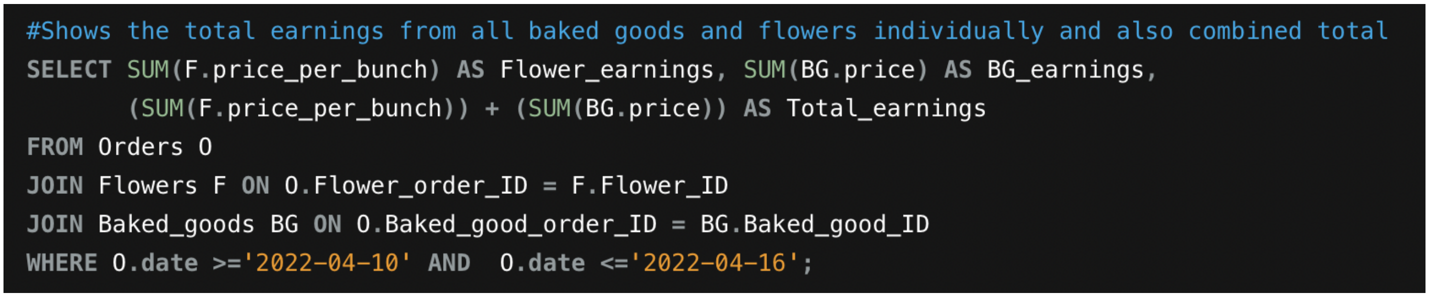
Name	Num_orders
Donna	3
Leslie	2
Ron	2
Winston	2
Aly	1
Anchal	1
Andy	1
Ann	1
April	1
Ben	1
Cece	1

…

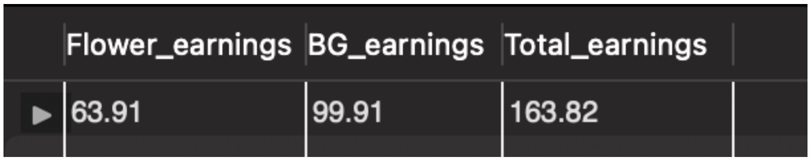


Cece	1
Chris	1
Jerry	1
Jess	1
Nick	1
Palak	1
Sandra	1
Schmidt	1
Tom	1
Varsha	1

#4: Show the cost of an occurrence, derived using aggregate functions

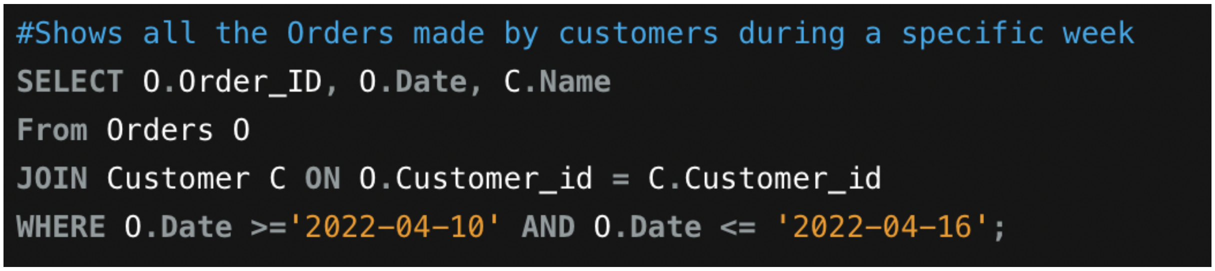


```
#Shows the total earnings from all baked goods and flowers individually and also combined total
SELECT SUM(F.price_per_bunch) AS Flower_earnings, SUM(BG.price) AS BG_earnings,
       (SUM(F.price_per_bunch)) + (SUM(BG.price)) AS Total_earnings
FROM Orders O
JOIN Flowers F ON O.Flower_order_ID = F.Flower_ID
JOIN Baked_goods BG ON O.Baked_good_order_ID = BG.Baked_good_ID
WHERE O.date >='2022-04-10' AND  O.date <='2022-04-16';
```

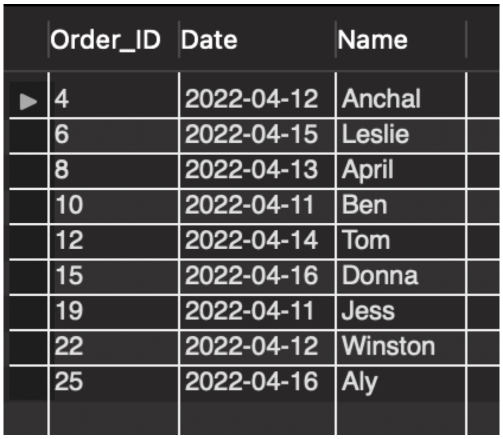


Flower_earnings	BG_earnings	Total_earnings
63.91	99.91	163.82

#5: Show a schedule for multiple occurrences, sorted by date and time



```
#Shows all the Orders made by customers during a specific week
SELECT O.Order_ID, O.Date, C.Name
From Orders O
JOIN Customer C ON O.Customer_id = C.Customer_id
WHERE O.Date >='2022-04-10' AND O.Date <= '2022-04-16';
```



Order_ID	Date	Name
4	2022-04-12	Anchal
6	2022-04-15	Leslie
8	2022-04-13	April
10	2022-04-11	Ben
12	2022-04-14	Tom
15	2022-04-16	Donna
19	2022-04-11	Jess
22	2022-04-12	Winston
25	2022-04-16	Aly

Implementation Log

[4/12/22 3:37PM] - Created flour&flowers.sql

- **CUSTOMER table** [4/12/22 3:52PM]
  - Customer_id TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
  - Name VARCHAR(255) NOT NULL,
  - C_Phone BIGINT(10) NOT NULL,
  - C_Email VARCHAR(255) NOT NULL,
  - Order_ID TINYINT(3) NOT NULL,
  - Payment VARCHAR NOT NULL,
  - CONSTRAINT orderID_fk_order
  - FOREIGN KEY (Order_ID)
  - REFERENCES Orders (Order_ID)

- **ORDERS table** [4/12/22 3:55PM]
  - Order_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
  - Flowers VARCHAR(255) NOT NULL,
  - Baked_goods VARCHAR(255) NOT NULL,
  - Date DATETIME NOT NULL,

- **FLOWERS table** [4/12/22 3:59 PM)
  - Flower_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
  - Flower_name VARCHAR(255) NOT NULL UNIQUE,
  - Price_per_bunch FLOAT(4 , 2) NOT NULL,
  - Color SET('Red', 'Pink', 'White') NOT NULL

- **BAKED_GOODS table** [4/12/22 4:02 PM)
  - Baked_good_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
  - Item_name VARCHAR(255) NOT NULL UNIQUE,
  - Recipe VARCHAR(255) NOT NULL,
  - Supplies SET('Flour', 'Sugar', 'Eggs'),
  - Price FLOAT(4 , 2) NOT NULL,
  - Best_before_date DATE NOT NULL
  -

- **PRODUCTS_OFFERED table** [4/12/22 4:06 PM)
  - Product_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
  - Baked_good_ID TINYINT(3),
  - Flower_ID TINYINT(3),
  - FOREIGN KEY (Baked_good_ID)
  - REFERENCES Baked_goods (Baked_good_ID),
  - FOREIGN KEY (Flower_ID)
  - REFERENCES Flowers (Flower_ID)

- **EMPLOYEE table** [4/12/22 4:09 PM)
  - Employee_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
  - First_name VARCHAR(255) NOT NULL,
  - Last_name VARCHAR(255) NOT NULL,

    - E_Email VARCHAR(255) NOT NULL,
    - E_Phone BIGINT(10) NOT NULL,
    - Manager_ID INT(5),
    - Salary INT(6) NOT NULL,
    - Job_Type SET('Manager', 'Baker', 'Packager') NOT NULL

- **SUPPLIER table** [4/12/22 4:10 PM)
    - Supplier_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Supplier_name VARCHAR(255) NOT NULL,
    - S_Phone_# BIGINT(10) NOT NULL,
    - S_Email VARCHAR(255) NOT NULL,
    - Supply_Type ENUM(‘Baking', 'Flower', 'Packaging') NOT NULL,

- **STORAGE_INVENTORY** [4/12/22 4:21 PM)
    - Supply_ID TINYINT(3) NOT NULL,
    - Supply_Name VARCHAR(255) NOT NULL,
    - Supply_Type SET('Baking', 'Flower', 'Packaging') NOT NULL,
    - Units_available INT NOT NULL,

4/14/22

- **BAKED_GOODS table** [4/12/22 12:57 PM)
    - Baked_good_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Item_name VARCHAR(255) NOT NULL UNIQUE,
    - Recipe VARCHAR(255) NOT NULL,
    - ~~Supplies SET('Flour', 'Sugar', 'Eggs'),~~
    - Price FLOAT(4 , 2) NOT NULL,
    - Best_before_date DATE NOT NULL

- **CUSTOMER table** [4/14/22 1:13PM]
    - Customer_id TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Name VARCHAR(255) NOT NULL,
    - C_Phone BIGINT(10) NOT NULL,
    - C_Email VARCHAR(255) NOT NULL,
    - ~~**Order_ID TINYINT(3) NOT NULL,**~~
    - ~~CONSTRAINT orderID_fk_order~~
    - ~~FOREIGN KEY (Order_ID)~~
    - ~~REFERENCES Orders (Order_ID)~~

- **SUPPLIER table** [4/12/22 4:10 PM)
    - Supplier_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Supplier_name VARCHAR(255) NOT NULL,
    - **S_Phone** BIGINT(10) NOT NULL,
    - S_Email VARCHAR(255) NOT NULL,
    - ~~Supply_Type ENUM(‘Baking', 'Flower', 'Packaging') NOT NULL,~~

4/17/22

- **FLOWERS table** [4/12/22 3:59 PM)
    - Flower_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Flower_name VARCHAR(255) NOT NULL UNIQUE,
    - Price_per_bunch FLOAT(4 , 2) NOT NULL,
    - Color SET('Red', 'Pink', 'White', **'Purple', 'Blue'**) NOT NULL

- **SUPPLIER table** [4/12/22 4:10 PM)
    - Supplier_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Supplier_name VARCHAR(255) NOT NULL,
    - S_Phone BIGINT(10) NOT NULL,
    - S_Email VARCHAR(255) NOT NULL,
    - ~~Supply_Type **SET**('Baking', 'Flower', 'Packaging') NOT NULL,~~

- **SUPPLIER table** [4/12/22 7:22 PM)
    - Supplier_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Supplier_name VARCHAR(255) NOT NULL,
    - S_Phone BIGINT(10) NOT NULL,
    - S_Email VARCHAR(255) NOT NULL,
    - ~~Supply_Type **SET**('Baking', 'Flower', 'Packaging') NOT NULL,~~
    - **Supply_ID TINYINT(3) NOT NULL,**
    - **FOREIGN KEY (Supply_ID)**
    - **REFERENCES Supplies (Supply_ID);**

4/27/22

- **CUSTOMER table** [4/27/22 2:17PM]
    - Customer_id TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Name VARCHAR(255) NOT NULL,
    - C_Phone BIGINT(10) NOT NULL,
    - C_Email VARCHAR(255) NOT NULL,
    - Order_ID TINYINT(3) NOT NULL,
    - ~~Payment VARCHAR NOT NULL,~~
    - CONSTRAINT orderID_fk_order
    - FOREIGN KEY (Order_ID)
    - REFERENCES Orders (Order_ID)

- **ORDERS table** [4/27/22 2:46PM]
    - Order_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Flowers VARCHAR(255) NOT NULL,
    - Baked_goods VARCHAR(255) NOT NULL,
    - Date DATETIME NOT NULL,
    - **Customer_id TINYINT(3) NOT NULL,**
    - **CONSTRAINT custID_fk_customer**
    - **FOREIGN KEY (Customer_ID)**
    - **REFERENCES Customer (Customer_id)**

- **EMPLOYEE table** [4/12/22 3:03 PM)
    - Employee_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - First_name VARCHAR(255) NOT NULL,
    - Last_name VARCHAR(255) NOT NULL,
    - E_Email VARCHAR(255) NOT NULL,
    - E_Phone BIGINT(10) NOT NULL,
    - Manager_ID INT(5),
    - Salary INT(6) NOT NULL,
    - Job_Type SET('Manager', 'Baker', 'Packager'**, 'Customer Service', 'Owner'**) NOT NULL

- **SUPPLIER table** [4/27/22 3:27 PM)
    - Supplier_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Supplier_name VARCHAR(255) NOT NULL,
    - S_Phone BIGINT(10) NOT NULL,
    - S_Email VARCHAR(255) NOT NULL,
    - Supply_ID TINYINT(3) NOT NULL,
    - **Store_Rep TINYINT(3), NOT NULL,**
    - FOREIGN KEY (Supply_ID)
    - REFERENCES Supplies (Supply_ID);
    - **FOREIGN KEY (Store_Rep)**
    - **REFERENCES Employee (Employee_ID)**

- **STORAGE_INVENTORY table** [4/27/22 4:21 PM)
    - **Inventory_item_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,**
    - Supply_ID TINYINT(3) NOT NULL,
    - Supply_Name VARCHAR(255) NOT NULL,
    - Supply_Type SET('Baking', 'Flower', 'Packaging') NOT NULL,
    - Units_available INT NOT NULL,
    - FOREIGN KEY (Supply_ID)
    - REFERENCES Supplies (Supply_ID)

- **SUPPLIES table** [4/27/22 4:35]
    - Supply_ID TINYINT(3) PRIMARY KEY AUTO_INCREMENT,
    - Supply_name VARCHAR(255) NOT NULL,
    - Supply_price FLOAT(4 , 2) NOT NULL,
    - Supply_type SET('Baking', 'Flower', 'Packaging') NOT NULL