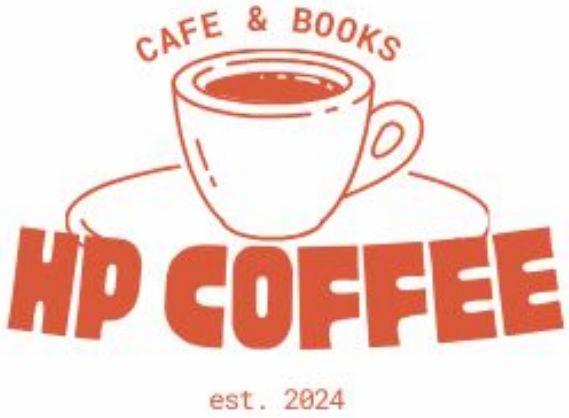
DATABASE FINAL PROJECT: STEP 3



Emma Griffin, Olivia Werba

Updated Description: Our goal for the final project was to create a database that simulates daily operations for a local cafe and bookstore and would allow for an optimization of workflows and efficiency. The database is modeled after Plein Air, who very graciously provided us with menus and invoices, which we used to populate our database.

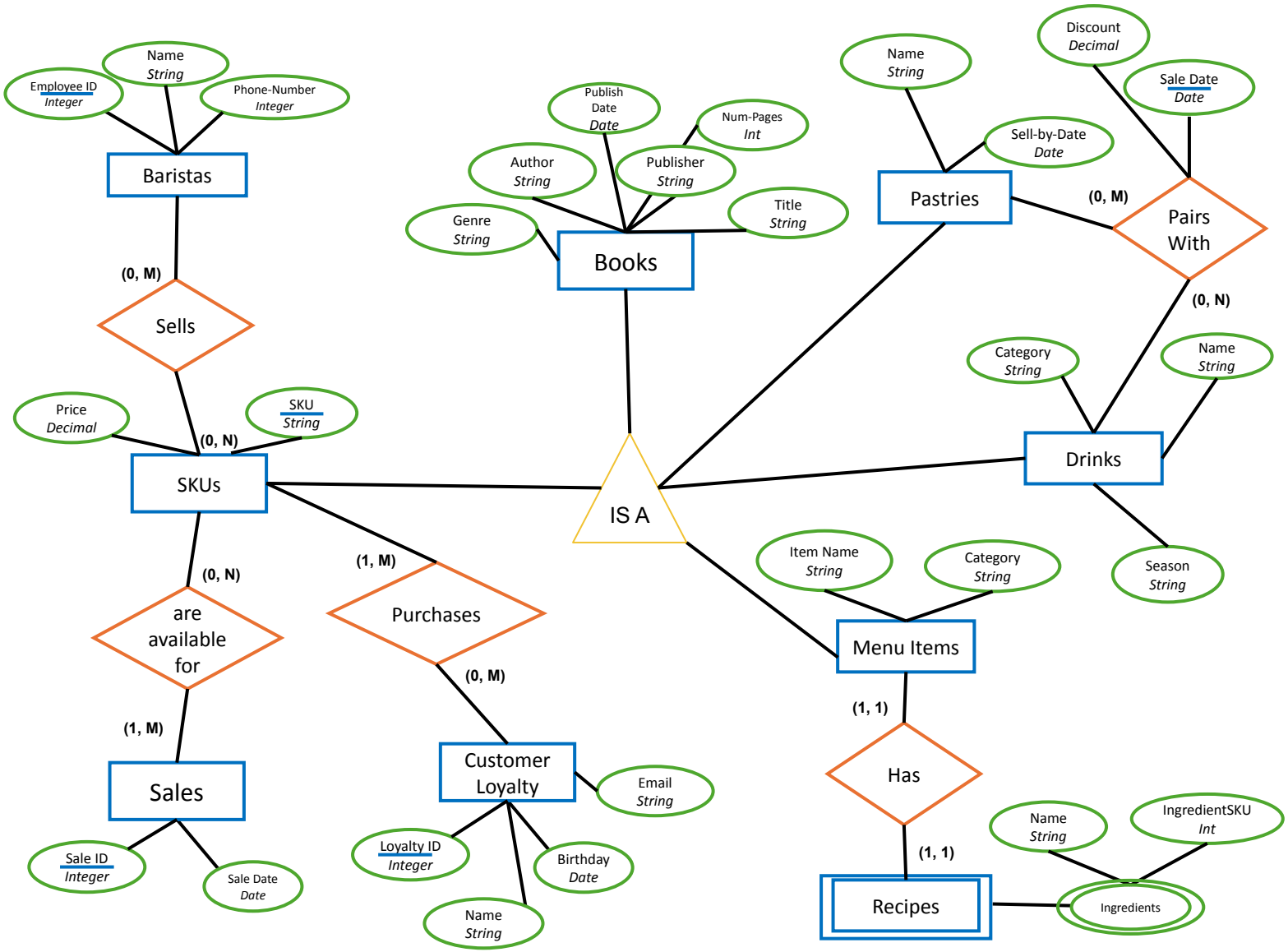
Updated Submitted Entity Sets:

1. Each item or good sold at HP Coffee has a 'SKU' and each is identified by the SKU number and also has a price.
2. Menu items (goods) also contain the item name, and the category.
3. Pastries (goods) also contain the pastry name and the sell-by-date.
4. Drinks (goods) also contain the drink name, category and season they are served.
5. Books (good) also contain the title, genre, author, publisher, date of publication, and number of pages.
6. Baristas are identified by their employee ID and should also contain their name and phone number.
7. Sales consist of the sale id and the sale date.
8. There is a customer loyalty program for frequent customers. Customer loyalty is identified by the customer loyalty ID and should also include the customers name, birthday, and email.

Updated Relationship Sets:

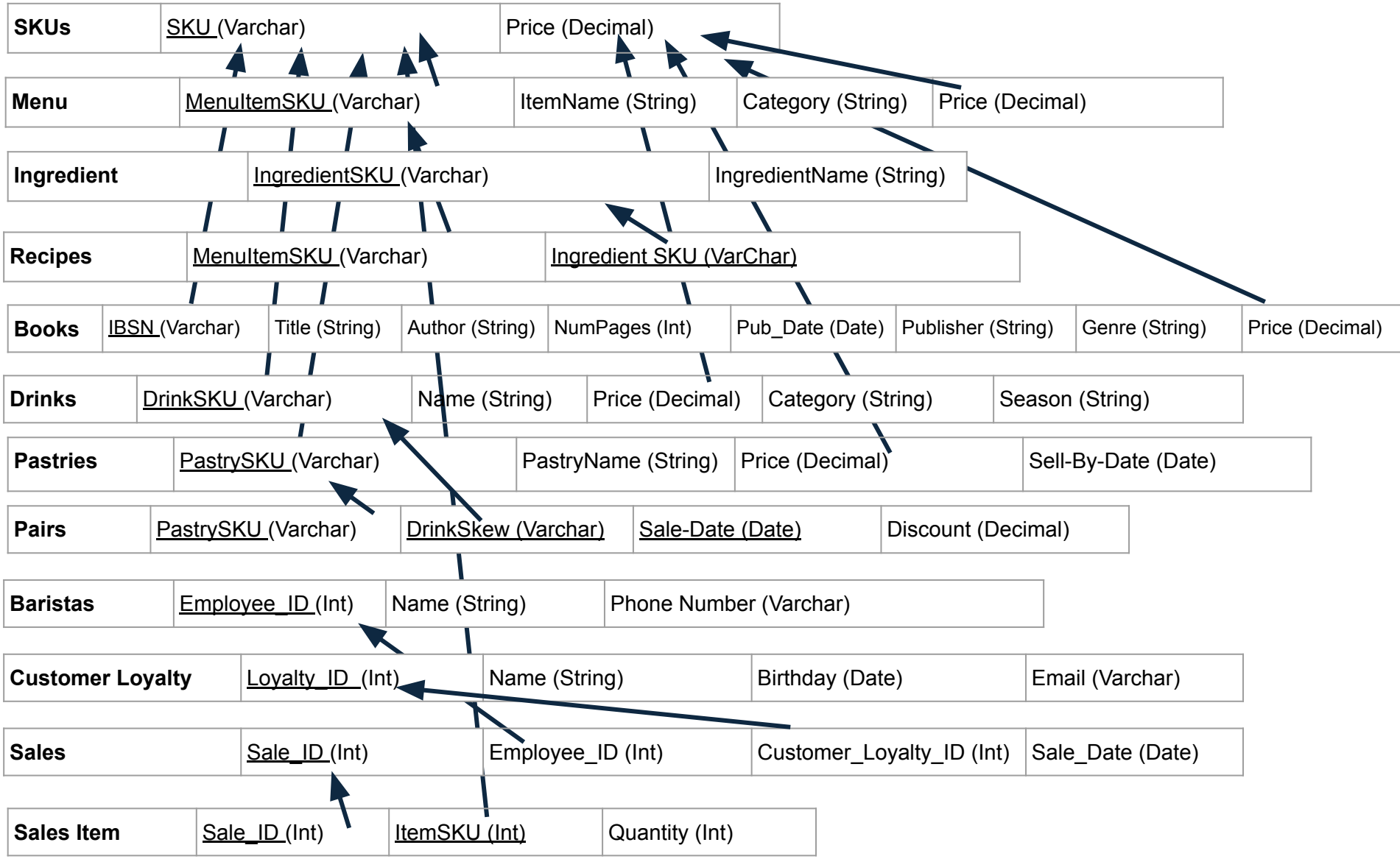
1. Multiple goods are sold by multiple baristas.
2. Many goods can be bought by many customers in the loyalty program.
3. A sale can consist of many goods, and many goods can be related back to a sale.
4. Each menu item has a recipe, which contains all the ingredients to make that item. If the menu item is removed, the recipe should be as well.
5. On certain dates, pastries and beverages are paired together with a discounted price. A beverage can pair with multiple pastries, and a pastry can pair with multiple beverages.

New Entity Sets, Relations, and ER Diagram:



Adjustments Made: We made sure to have goods identified by unique SKUs, rather than names. We also reorganized the 'IS A' superclass to include books, allowing for queries to retrieve all goods sold.

* The ‘SKU’ attribute of goods is a string to account for the ISBN of books containing letters



Adjustments Made: The relational schema was adjusted to match the new ER diagram. When we began the physical database design step of our project, we de-normalized our tables to include the 'price' attribute in the various 'goods' tables, because we were querying frequently for this information, and joining 'SKUs' every time seemed unnecessary.

We also combined the relationships pertaining to 'sale' into one table that included barista, customer, and sales information, and the items that were sold into another table. Again, this was to reduce the amount of joins and make the database more efficient.

Data:

In our dataset, the **Books** table minus the price column, is entirely populated by real data from this kaggle dataset.

The **Menu** table is also populated with real data scraped from the Plein Aire website. Because of this, the ingredient table is also populated with real data, as it was generated from the descriptions provided on the website.

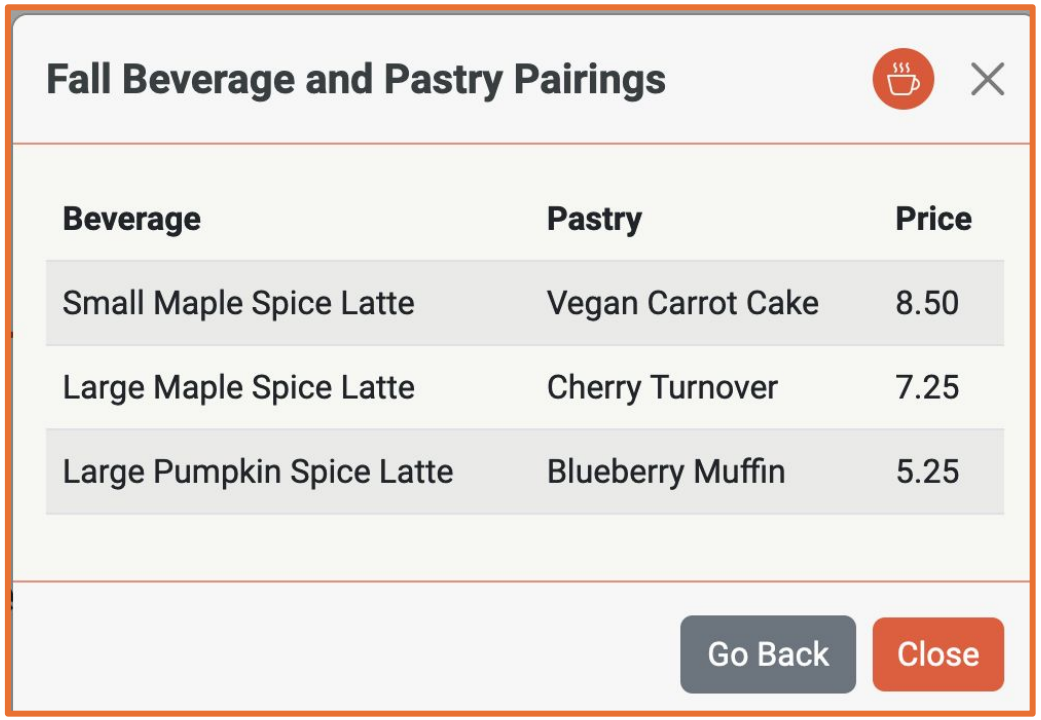
The **Pastry and Drinks** table are a blend of real and generated data. For pastries, items such as *Baguette Tradition*, *Plain Croissant*, *Chocolate Croissant*, *Almond Croissant*, *Beignet Hazelnut Filled*, *Long Bread "La Fournette"*, *Pain de Mie*, *Pistachio Macaron*, *Chocolate Macaron*, and *Mango/Passionfruit Macaron*, along with their prices, were sourced from a Plein Air pastry invoice

Similarly, the **Drinks** table includes real Plein Air drink items, along with their prices and categories, but has been expanded to provide additional variety for the database.The original drinks include *Espresso, Macchiato, Cortado, Cappuccino, Latte Mocha, Americano, Drip Coffee, Pour Over, Premium Pour Over, Cold Brew, Premium Cold Brew, Loose Leaf Tea, Assorted Iced Tea, Assorted Homemade Chai, Hot Chocolate, and Steamed Milk.*

QUERY 1

Fall Beverage and Pastry Pairings with Their Discounted Price:

```
SELECT D.Item AS Beverage, P.PastryName AS Pastry, (D.Price  + P.Price - Pair.Discount) AS
Price
FROM Drinks AS D
JOIN Pairs AS Pair ON D.DrinkSKU  = Pair.DrinkSKU
JOIN Pastries AS P ON Pair.PastrySKU  = P.PastrySKU
WHERE D.Season  = 'Fall'
ORDER BY Price DESC ;
```



QUERY 2

Breakfast Items Sold, Sorted by Amount Sold:

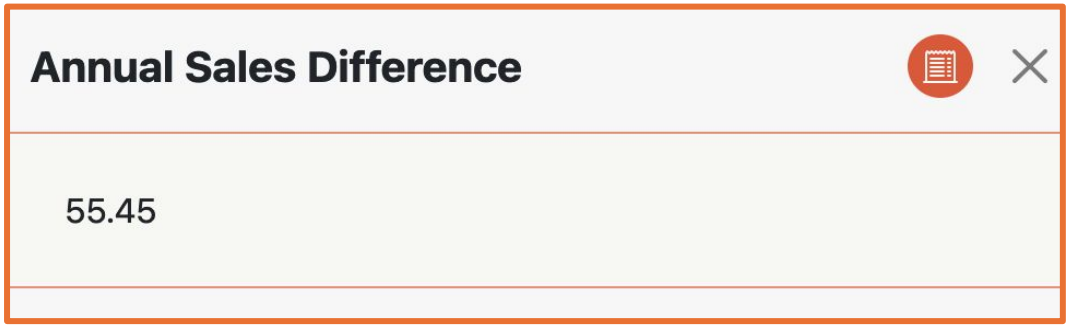
```
SELECT
    M.ItemName AS ITEM,
    SUM(SI.Quantity) AS AMTSOLD
FROM
    Sales_Item AS SI
JOIN
    Menu AS M ON SI.Item = M.MenuItemSku
WHERE
    M.Category = 'Breakfast'
GROUP BY
    M.ItemName
ORDER BY
    AMTSOLD DESC
;
```

Popular Breakfast Items

AMTSOLD	ITEM
18	Plein Continental Breakfast
18	Chorizo Burrito
12	Plein Omelet
11	House Made Biscuit Sandwich
10	Breakfast Croissant Sandwich
9	Two Eggs Any Style
9	Mascarpone Stuffed Brioche French Toast
9	Veggie Burrito

QUERY 3 Annual Sales Difference Between 2023 and 2022:

```
SELECT
    (SELECT
        SUM(SKEW.Price * SI.Quantity)
    FROM
        Sales_Item AS SI
    JOIN
        Skews AS SKEW ON SI.Item = SKEW.Skew
    JOIN
        Sales AS SALES ON SALES.Sale_ID = SI.Sale_ID
    WHERE
        YEAR(SALES.Sale_Date) = 2023)
    -
    (SELECT
        SUM(SKEW.Price * SI.Quantity)
    FROM
        Sales_Item AS SI
    JOIN
        Skews AS SKEW ON SI.Item = SKEW.Skew
    JOIN
        Sales AS SALES ON SALES.Sale_ID = SI.Sale_ID
    WHERE
        YEAR(SALES.Sale_Date) = 2022) AS SalesDifference;
```



QUERY 4

Baristas Total Sales, Sorted by Sales Amount:

```
SELECT B.Name AS Name, SUM(SKU.Price * SI.Quantity) AS TotalSales
FROM Skews AS SKU
JOIN Sales_Item AS SI ON SI.Item = SKU.Skew
JOIN Sales AS S on S.Sale_ID = SI.Sale_ID
JOIN Baristas AS B ON B.Employee_ID = S.Employee_ID
GROUP BY B.Employee_ID
ORDER BY TotalSales DESC;
```

Star Employees

Name	TotalSales
William Parker	2378.17
Charlotte Hall	2377.08
Ashley Carter	2304.56
Ava Turner	2252.47
Isabella Scott	2213.90
Violet Brooks	2152.03
Mia Rodriguez	2151.61
Liam Harris	2078.61
Sophia Mitchell	2070.20
Ethan Bennett	1891.63

QUERY 5

Book Categories Sold, Sorted by Amount Sold:

```
SELECT
   B.Genre, COUNT(SI.Item) as COUNT
FROM
   Books as B
JOIN
   Sales_Item AS SI ON SI.Item = B.ISBN
GROUP BY
   B.Genre
ORDER BY
   COUNT DESC;
```

Popular Book Genres

COUNT	Genre
67	Fiction
58	Classics
47	Fantasy
32	History
28	Nonfiction
25	Mystery
22	Sequential Art
19	Science Fiction
12	Childrens
12	Philosophy

QUERY 6

Barista ‘Vegan Carrot Cake’ Sales, Sorted by Amount Sold:

```
SELECT B.Name AS Name, B.Phone AS PhoneNumber,
COUNT(SI.Quantity) AS CakeSold
FROM Pastries AS P
JOIN Sales_Item AS SI ON SI.Item  = P.PastrySkew
JOIN Sales AS S on S.Sale_ID  = SI.Sale_ID
JOIN Baristas AS B ON B.Employee_ID  = S.Employee_ID
WHERE P.PastryName  = 'Vegan Carrot Cake'
GROUP BY B.Name, B.Phone
ORDER BY CakeSold DESC;
```

Cake Sales

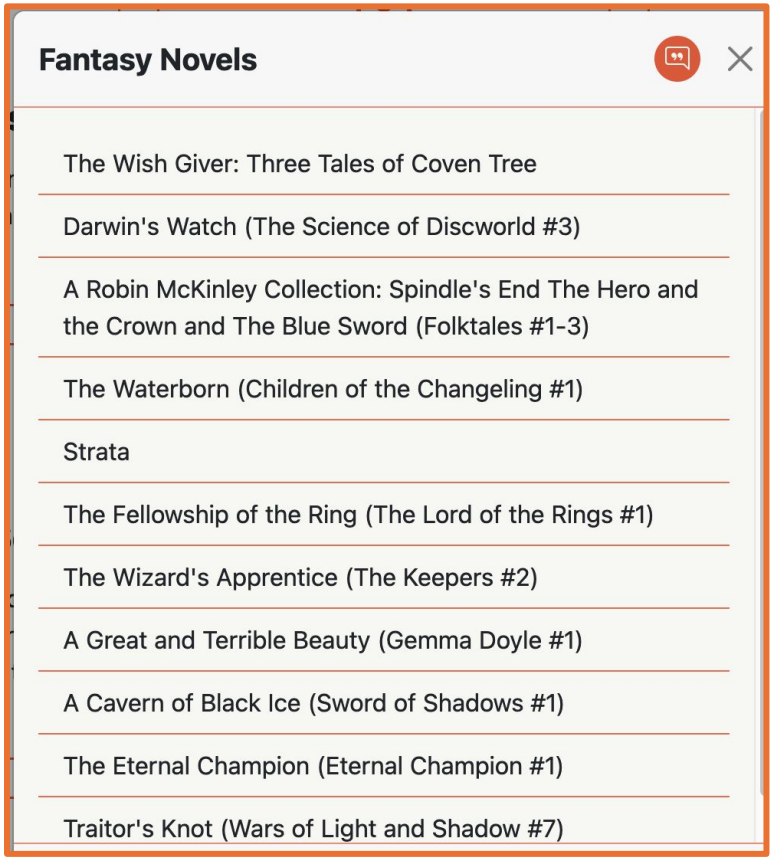
CakeSold	Name	PhoneNumber
3	Isabella Scott	312-901-2345
3	Mia Rodriguez	312-345-6789
3	Sophia Mitchell	312-567-8901
2	Mason Wright	312-567-8903
2	Ashley Carter	312-123-4567
2	Liam Harris	312-678-9012
2	Lucas Evans	312-345-6781
2	Ethan Bennett	312-234-5678
1	William Parker	312-890-1234
1	Noah Walker	312-456-7890

QUERY 7

Fantasy Books Sold in a User Given Year:

```
SELECT B.Title
FROM Books as B
JOIN Skews AS S ON S.Skew = B.ISBN
JOIN Sales_Item AS SI on SI.Item = S.Skew
JOIN Sales AS SAL on SAL.Sale_ID =
SI.Sale_ID
WHERE B. Genre = 'Fantasy' AND
YEAR(SAL.Sale_Date) = %s;
```

(query results shown with year 2023)



QUERY 8

Most Common Ingredients and the Maximum Price of the Item That Contains That Ingredient, Sorted by Number of Menu Items Using That Ingredient:

```
SELECT I.IngredientName AS Ingredient, COUNT(R.IngredientSkew) AS
NumberOfMenuItems, Max(M.Price) AS MaxPriceMenuItem
FROM Recipes AS R
JOIN Menu AS M ON R.MenuItemSkew = M.MenuItemSkew
JOIN Ingredients AS I ON R.IngredientSkew = I.IngredientSkew
GROUP BY I.IngredientName
HAVING COUNT(R.IngredientSkew) > 3
ORDER BY COUNT(R.IngredientSkew) DESC;
```

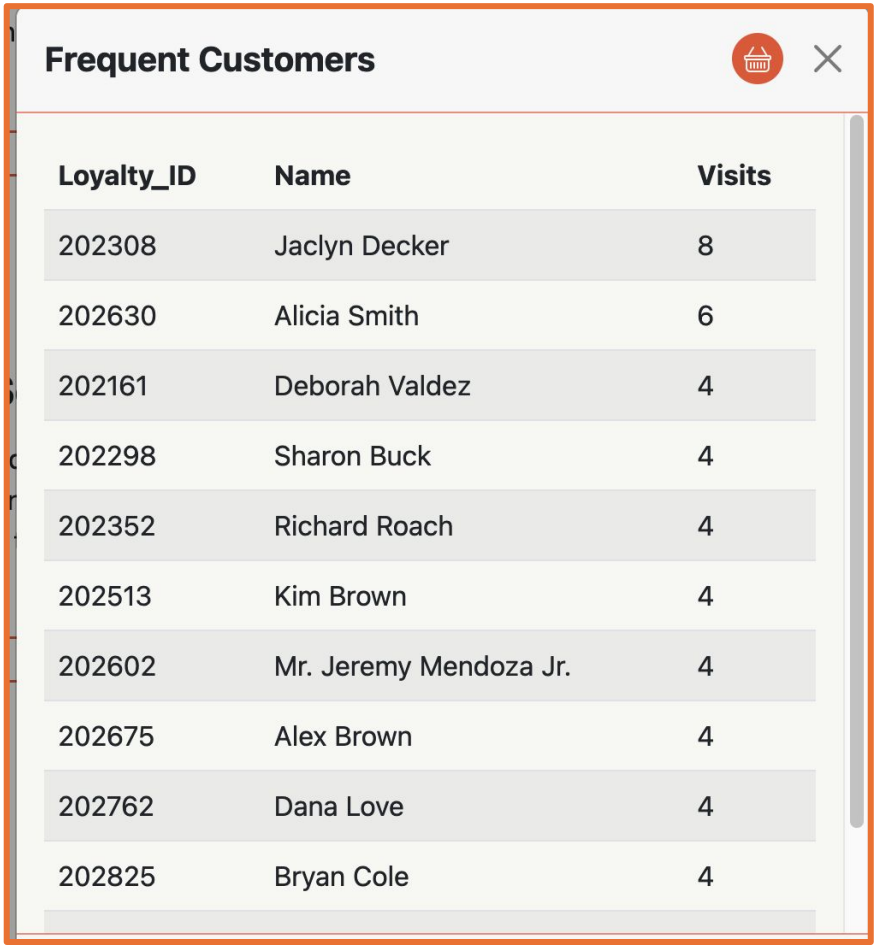
Common Ingredients

Ingredient	MaxPriceMenuItem	NumberOfMenuItems
arugula	12.50	7
baguette	14.95	6
manchego cheese	14.50	5
avocado	14.95	5
egg	11.95	4
home fries	11.95	4

QUERY 9

Customers in the Loyalty Program Who Have Visited More Than 3 Times, Sorted by Total Visits:

```
SELECT CL.Name, CL.Loyalty_ID, COUNT(S.Sale_ID) AS Visits
FROM Customer_Loyalty as CL
JOIN Sales AS S ON S.Customer_Loyalty_ID = CL.Loyalty_ID
GROUP BY CL.Loyalty_ID
HAVING Visits > 3
ORDER BY Visits DESC;
```



Frequent Customers

Loyalty_ID	Name	Visits
202308	Jaclyn Decker	8
202630	Alicia Smith	6
202161	Deborah Valdez	4
202298	Sharon Buck	4
202352	Richard Roach	4
202513	Kim Brown	4
202602	Mr. Jeremy Mendoza Jr.	4
202675	Alex Brown	4
202762	Dana Love	4
202825	Bryan Cole	4

QUERY 10

Total Items, Quantity Sold, and Total Sales in Each Menu Item Category:

```
SELECT M.Category, COUNT(M.MenuItemSkew) AS MenuItemCount, SUM(SI.Quantity) As
QuantitySold, SUM(M.Price * SI.Quantity) AS TotalSales
FROM Menu AS M
JOIN Sales_Item AS SI ON SI.Item = M.MenuItemSkew
GROUP BY M.Category
ORDER BY M.Category;
```

Menu Items by Category

Category	MenuItemCount	QuantitySold	TotalSales
Breakfast	54	96	1159.70
Salads, Sandwiches & Baguettes	87	160	1872.30
Tartines & Plein Bowls	21	46	483.00

Thank You!