0. Preface

I am writing this section to both apologize and explain the circumstances surrounding my performance in the course and my project. Over the course of this semester, I had some personal challenges including two significant family issues, which impacted my mental health and my overall ability to perform academically as I normally do. Unfortunately I let these issues cause me to miss several of your classes (along with some of my other classes as well), which significantly impacted my understanding and progress in the course. I recognize this was a critical time, and missing these sessions put me at a disadvantage, not only in keeping up with the coursework but also in managing the project effectively. I understand the importance of staying engaged with the curriculum and regret not communicating these issues sooner.

My communication was not as proactive as it should have been. I understand that I should have reached out sooner for guidance and assistance, especially after learning that my partner had withdrawn from the class, leaving me to complete our project independently. This was an additional challenge, as we had only really chosen a store and started brainstorming tables and things like that.

Despite these setbacks, I have completed the project to the best of my ability under the circumstances / time frame. I have learned a valuable lesson about the importance of communication and managing unforeseen challenges. It does make me sad thinking about this because I really loved your course along with your teaching style. I really feel like this semester doesn't accurately represent who I am as a student. I sincerely apologize for any inconvenience my situation may have caused the class and appreciate any feedback you have regarding the project. Thank you for your understanding.

1. Description of the Store, Products, Customers, etc.

Michaels is a privately owned chain of American / Canadian arts and crafts stores. It is North America's largest provider of arts, crafts, framing, floral and wall decor, and more. The company is part of The Michaels Companies, founded in 2014 and headquartered in Texas. In addition to Michaels stores, The Michaels Companies operates Aaron Brothers Custom Framing store-within-a-store, and Artistree, a manufacturer of custom / specialty framing merchandise.

Michaels sell a variety of arts and crafts products, including scrapbooking, beading, knitting, rubber stamping, home décor items, floral items, kids' crafts, paints, framing, greenery, baking, and many seasonal items. Online, customers can upload, edit and print photos from their computer or social media sites. They can then select and customize the size, surface, frame, and mat. As of January 2021, there were 1,252 Michaels stores in 49 out of 50 states in the U.S., and Canada, with approximately $5.362 billion in sales for 2020.

For this project, my version of the Michaels Craft Store operates both physical and online retail outlets. The stores offer a variety of products ranging from basic craft supplies to specialized art materials. Products are organized into various categories such as art supplies, sewing supplies, and painting supplies, with more specific subcategories like colored pencils, fabric scissors, and acrylic paints. These products come from brands like Crayola, Fiskars, and Winsor & Newton, ensuring quality and customer satisfaction.

My database design incorporates the following specific business rules:

- Each product is associated with a brand and falls under a specific product type.
- Prices can vary by store location to accommodate regional market conditions.
- Inventory levels are tracked per store to facilitate efficient stock management.
- Customers can shop online or at physical store locations, and their interactions (purchases and cart activities) are stored to enhance the shopping experience.
- Vendors supply specific brands, and this relationship is managed to streamline procurement.
- Vendors ship their products directly to the stores.

Overview:

- The database is designed to manage a chain of craft stores, including an online store. Physical stores are located in various locations. The online store serves customers nationwide without a physical storefront, operating 24/7.

Products:

- Products include art supplies, sewing supplies, painting supplies, and more, from brands like Crayola, Fiskars, Loops & Threads, and Winsor & Newton. Each product has specific attributes such as UPC, brand, and type.

Customers:

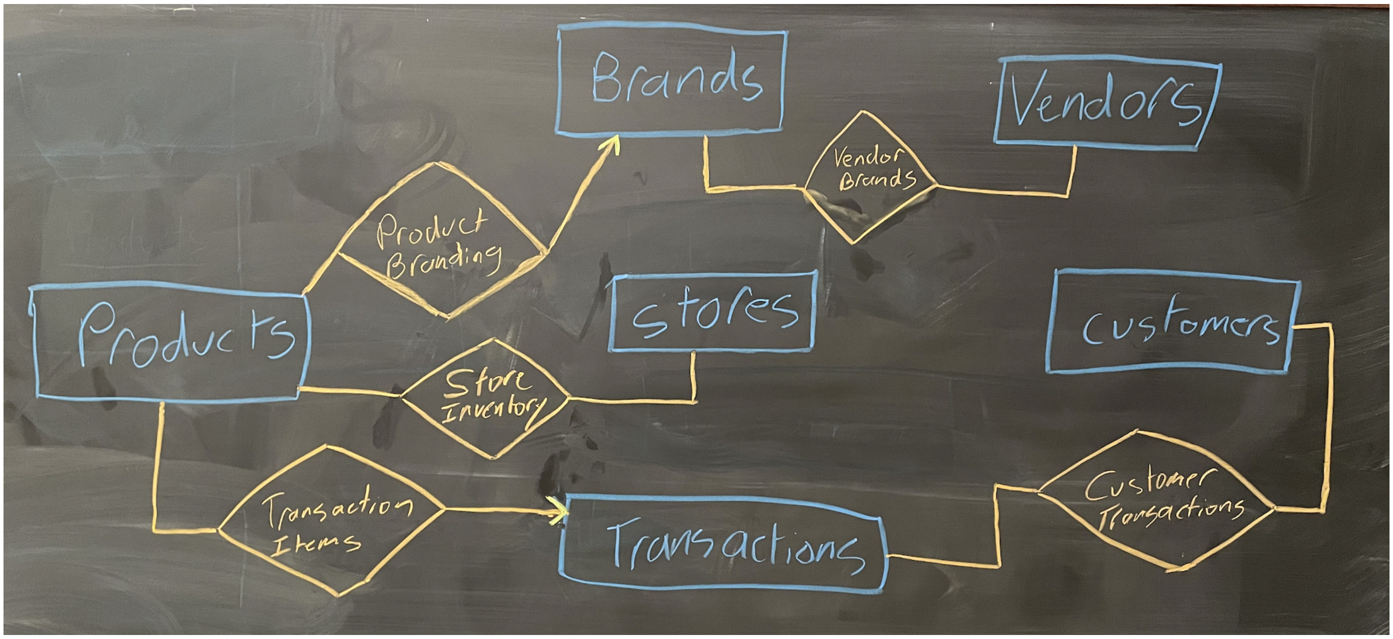
- Customers can interact with the system by adding products to their shopping carts, placing orders, and reviewing their past orders.

2. Significant Problems and Solutions

Throughout the development of the database, I encountered several challenges:

- **Schema Adjustments for Realistic Business Operations**: Initially, the schema did not account for multiple products in a single transaction effectively. This was resolved by introducing a TransactionItems table, which is a many-to-many relationship. Although, at first, I thought it was a one-to-many relationship since it's like one singular transaction can have multiple products but really it's many-to-many since different transactions can have the same products. This allowed me to handle multiple products per transaction and accurately track inventory and sales.
- **Dynamic Pricing and Inventory Management**: To address pricing flexibility across different regions and inventory management at multiple store locations, I implemented ProductPricing and Inventory tables that relate products to different stores with specific prices and stock levels.
- **User Authentication and Role Management**: To facilitate different user interactions with the database, I expanded the Users table to include roles and associated permissions, enhancing security and user experience.
    - Problem: Needed a secure way to handle different types of users (managers, vendors, customers, DBAs) with varying access levels.
    - Solution: Implemented a Users table with roles and optional links to vendors, enabling role-based access control within the system.
    - Customers however, are not included since they have their own table and also their own separate authentication process.

3. E-R Diagram and Explanatory Notes



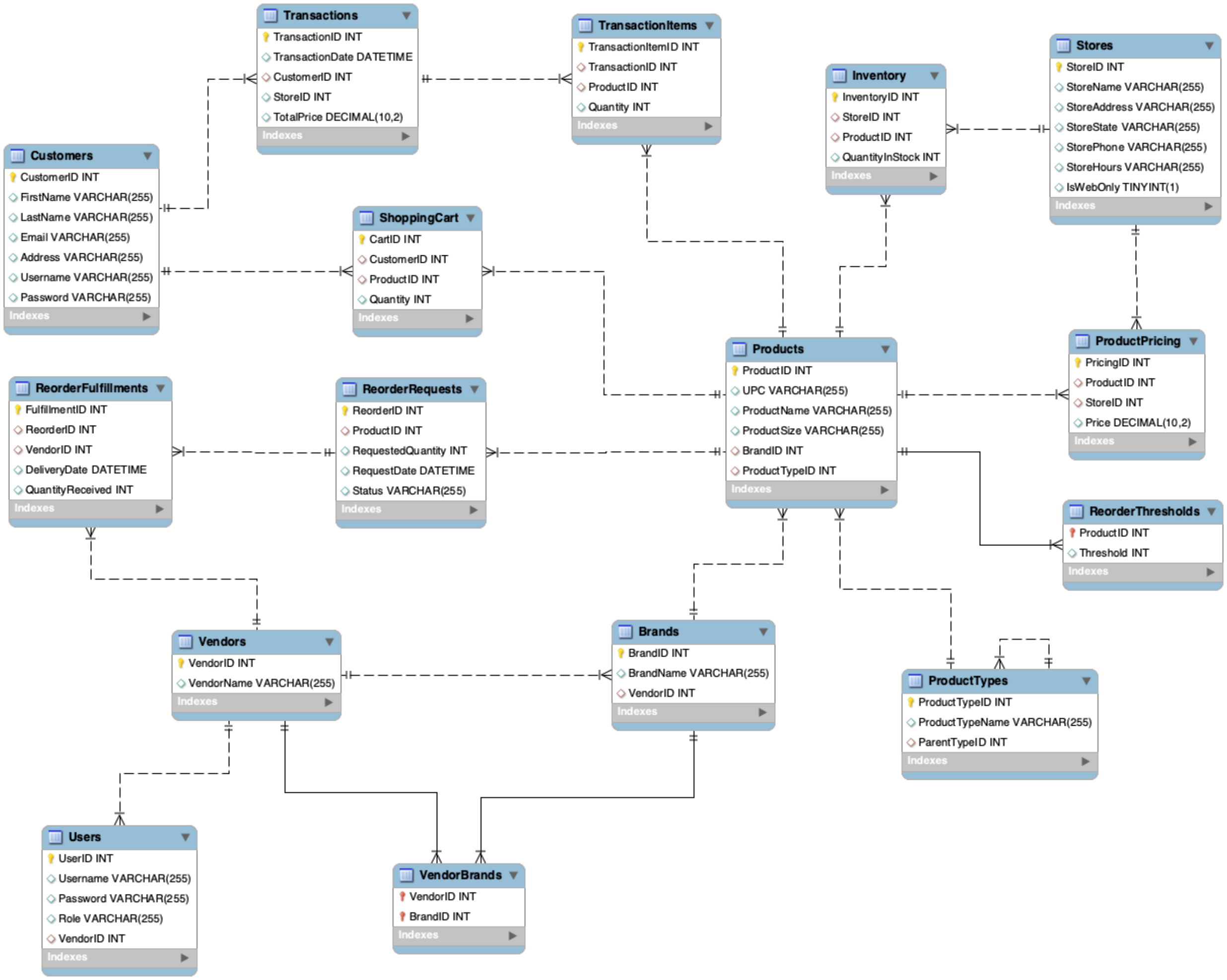
Entities:

1. **Customers:** Represents the customers who purchase products.
2. **Stores:** Represents physical and online store locations.
3. **Products:** Represents items sold in the stores.
4. **Brands:** Represents the brands of products sold.
5. **ProductTypes:** Represents categories of products.
6. **Vendors:** Represents suppliers of products.
7. **Transactions:** Represents sales transactions.

Relationships:

8. **Product Branding:** Connects products to brands as a many-to-one relationship as many products can belong to one brand.
9. **Store Inventory:** Associates products with stores where they are sold. This is a many-to-many relationship as stores carry multiple products and products can be found in multiple stores.
10. **Customer Transactions:** Links transactions to customers who made the purchases. This is a one-to-many relationship as a customer can have many transactions.
11. **Transaction Items:** Details products and quantities in a transaction. This is a many-to-many relationship between products and transactions.
12. **Vendor Brands:** Represents which brands are supplied by which vendors. This is a many-to-many relationship as vendors can supply multiple brands and brands can be supplied by multiple vendors.

4. Relational Schema



Entities:

- Users
- Vendors
- Brands
- ProductTypes
- Stores

- Products
- ProductPricing
- Inventory
- Customers
- ShoppingCart
- Transactions
- TransactionItems
- VendorBrands (Junction table for many-to-many relationship)
- ReorderThresholds
- ReorderRequests
- ReorderFulfillments

Relationships

- Users and Vendors
  - One-to-many from Vendors to Users
- Vendors and Brands
  - One-to-many from Vendors to Brands
- ProductTypes (Self-referencing because of subcategories)
  - One-to-many hierarchical relationship within ProductTypes
- Products and Brands
  - One-to-many from Brands to Products
- Products and ProductTypes
  - One-to-many from ProductTypes to Products
- ProductPricing
  - Many-to-many between Products and Stores managed via ProductPricing table

- Inventory
    - Many-to-many between Products and Stores managed via Inventory table
- ShoppingCart
    - Many-to-many between Customers and Products managed via ShoppingCart table
- Transactions
    - One-to-many from Customers to Transactions
- TransactionItems
    - Many-to-many between Transactions and Products managed via TransactionItems table
- VendorBrands
    - Many-to-many between Vendors and Brands managed via VendorBrands table.
- ReorderThresholds
    - One-to-one relationship between Products and ReorderThresholds
- ReorderRequests
    - One-to-many from Products to ReorderRequests
- ReorderFulfillments
    - Many-to-one from ReorderFulfillments to ReorderRequests
    - Many-to-one from ReorderFulfillments to Vendors

5. Sample Queries

```sql
-- Query to find all products under $10 available online

SELECT ProductName, Price FROM Products

JOIN ProductPricing ON Products.ProductID = ProductPricing.ProductID

WHERE Price < 10 AND StoreID = (SELECT StoreID FROM Stores WHERE IsWebOnly = TRUE);
```

```sql
-- Query to check inventory levels below threshold

SELECT ProductName, QuantityInStock FROM Products

JOIN Inventory ON Products.ProductID = Inventory.ProductID

WHERE QuantityInStock < (SELECT Threshold FROM ReorderThresholds WHERE ProductID = Products.ProductID);
```

```sql
-- Query to get details of all transactions made by a specific customer

SELECT TransactionDate, ProductName, Quantity, TotalPrice FROM Transactions

JOIN TransactionItems ON Transactions.TransactionID = TransactionItems.TransactionID

JOIN Products ON TransactionItems.ProductID = Products.ProductID

WHERE CustomerID = (SELECT CustomerID FROM Customers WHERE Username = 'emily');
```

```
# What are the 20 top-selling products at each store?
    SELECT S.StoreName, P.ProductName, SUM(TI.Quantity) as TotalSold
    FROM TransactionItems TI
    JOIN Transactions T ON TI.TransactionID = T.TransactionID
    JOIN Products P ON TI.ProductID = P.ProductID
    JOIN Stores S ON T.StoreID = S.StoreID
    GROUP BY S.StoreName, P.ProductName
    ORDER BY S.StoreName, TotalSold DESC
    LIMIT 20;
```

```
# What are the 20 top-selling products in each state?
    SELECT S.StoreState AS State, P.ProductName, SUM(TI.Quantity) as TotalSold
    FROM TransactionItems TI
    JOIN Transactions T ON TI.TransactionID = T.TransactionID
    JOIN Products P ON TI.ProductID = P.ProductID
    JOIN Stores S ON T.StoreID = S.StoreID
    GROUP BY S.StoreState, P.ProductName
    ORDER BY S.StoreState, TotalSold DESC
    LIMIT 20;
```

```
# What are the 5 stores with the most sales so far this year?

    SELECT S.StoreName, SUM(TI.Quantity) as TotalSales
    FROM Transactions T
    JOIN TransactionItems TI ON T.TransactionID = TI.TransactionID
    JOIN Stores S ON T.StoreID = S.StoreID
    WHERE YEAR(T.TransactionDate) = YEAR(CURDATE())
    GROUP BY S.StoreName
    ORDER BY TotalSales DESC
    LIMIT 5;


# In how many stores does Crayola outsell Fiskars?

    SELECT S.StoreName,
        SUM(CASE WHEN B.BrandName = 'Crayola' THEN TI.Quantity ELSE 0 END)
        AS CrayolaSales,
        SUM(CASE WHEN B.BrandName = 'Fiskars' THEN TI.Quantity ELSE 0 END)
        AS FiskarsSales
    FROM TransactionItems TI
    JOIN Transactions T ON TI.TransactionID = T.TransactionID
    JOIN Products P ON TI.ProductID = P.ProductID
    JOIN Brands B ON P.BrandID = B.BrandID
    JOIN Stores S ON T.StoreID = S.StoreID
    GROUP BY S.StoreName
    HAVING CrayolaSales > FiskarsSales
```

```
# What are the top 3 types of product that customers buy in addition to Crayola Watercolor Set?
        SELECT PT.ProductTypeName, SUM(TI.Quantity) AS TotalQuantity
        FROM TransactionItems TI
        JOIN Transactions T ON TI.TransactionID = T.TransactionID
        JOIN Products P ON TI.ProductID = P.ProductID
        JOIN ProductTypes PT ON P.ProductTypeID = PT.ProductTypeID
        WHERE T.TransactionID IN (
                SELECT DISTINCT TI2.TransactionID
                FROM TransactionItems TI2
                JOIN Products P2 ON TI2.ProductID = P2.ProductID
                WHERE P2.ProductName = 'Crayola Watercolor Set'
        )
        AND P.ProductName != 'Crayola Watercolor Set'
        GROUP BY PT.ProductTypeID
        ORDER BY TotalQuantity DESC
        LIMIT 3;
```

6. Description of the Interface

Overview

In developing the Python based interface for the Michaels craft store database, I aimed to provide a structured and interactive environment tailored to the specific needs of different types of users: managers, customers, vendors, and database administrators (DBAs). Each user category accesses the database through a unique set of functionalities, enabling efficient data management, transaction processing, inventory control, and enhanced user interaction. The interface facilitates these interactions via the command-line interface, ensuring simplicity and accessibility for all users.

Structure and Design

The program's architecture includes several classes, each taking care of the functionalities necessary for each specific role within the system. The foundational class, **DatabaseConnector**, handles all aspects of database connectivity using MySQL, including user authentication based on their roles. Derived from this, the **ManagerInterface** offers tools for managers to access sales data, monitor inventory levels, and execute managerial queries critical for store operations. The **CustomerInterface** allows customers to interact seamlessly with the database, whether it’s viewing products, adding items to their shopping cart, placing orders, or reviewing past purchases, ensuring a smooth shopping experience. Meanwhile, the **VendorInterface** is crucial for maintaining supply chain efficiency, allowing vendors to manage reorder requests, confirm shipments, and view order fulfillment statuses. Lastly, the **DBAInterface** provides database administrators with the capabilities to execute SQL queries directly, which is vital for database maintenance and complex data manipulations.

Development Process

The development process was highly iterative, starting with the gathering of requirements and followed by the sequential design, coding, and testing of each component. A key focus was on creating a modular and maintainable code structure where each class inherits from the DatabaseConnector to streamline interactions with the database and reduce redundancy. The development was punctuated by a strong emphasis on handling exceptions and ensuring the reliability of database connections, essential for the robustness of the application.

Encountered Issues and Resolutions

Throughout the development, several challenges emerged. Managing different user roles initially proved to be difficult, which I addressed by refining the authentication process to incorporate role-based access control, ensuring users could only access functionalities pertinent to their permissions. Inventory management also presented difficulties, particularly in reflecting real-time updates post-shipment confirmations by vendors. This issue was resolved by adjusting how transactions were handled in the database, ensuring inventory levels were promptly updated. Enhancing error handling was another critical area of focus, where I implemented comprehensive checks and balances, especially around database transactions and user inputs, to improve the application's stability. Lastly, to enhance user experience, especially for non-technical users such as customers and managers, I focused on creating clear and intuitive navigation within the program, refining command-line inputs, and providing clear instructions and feedback throughout user interactions.