INF 1343 Data Modelling and Database Design
Assignment 1: Data Modelling

by
Kuan Yi Chou(1003850674)
Ruolan Zhang(1004711010)
Yiran Zheng(1003993725)

Master of Information, Faculty of Information
Course Instructor: Maher Elshakankiri
March 6th, 2023

Table of content

1. Executive Summary:

In this report, we examine the structure of the xxx company through different database designs. Through interviews with the boss and an employee and research on a corporate website, our team is able to navigate through the life cycle journey of customers visiting the website. Although we tried to be thorough in our investigation process, we still had to make some assumptions regarding item policy or consumer activities. Our main assumptions are that the boss does not participate in daily work, the philosophy of being a customer and a couple particular payment cases.

We were able to demonstrate the flow of the company through three different diagrams. The ERD, the EERD and the UML diagram. We then compared the strength and weakness of the diagrams and in terms of several real world use cases. Finally, we provided everyone with the data catalog documenting each of the individual attribute and entity types that were mentioned in the study.

Although we are unable to provide the actual name and information of the company that we included in the study, we hope that the information we gathered can help describe the structure of entities within the company well. We also want to make sure that we were able to describe the relationship between entities clearly. Finally, we want to describe each entity with attributes clearly.

2. Context for the Study:

The organization that we studied is an online store that sells toys, stationery, homeware, tableware and gifts. It is a one-stop-shop for all things and contains a range of products from different brands. Customers can browse the product in various colors, styles, and price ranges. Choose from different sizes, styles, and designs to suit their individual needs and preferences. The website provides a seamless shopping experience, with easy-to-use search and browse functions, clear product descriptions, and high-quality product images. Customers can add shopping carts and order products through online payment, and have them delivered directly to their doorstep. Customer service (Employee) will update the stock and pack the products upon receipt of the order, and keep track of the status of the customer's order after it has been shipped. When customers need after-sales service or other needs, employees will respond.

The organization is a small business, which means that there are no specific leading and executive departments aside from the boss. The boss usually handles the human resources department issues. The boss hires employees that handle the day to day transactions, interaction with the customers and the employees will be responsible to handle restock requests as well.

Although we have conducted our interview with the owner, there are still some aspects of which we are not certain. Thus we made assumptions about, the first one is that the boss does not participate at all with customers. We have made another assumption that the definition of customer does not mean that they have to make a purchase, but just need to register their information with the company. Additionally, we assume that individuals who purchase through gift cards would still have to input their information to receive the product. So anyone who creates an account OR purchase through gift cards and such would be a registered customer in our scenario.

The items sold are divided into some of their unique attributes that we feel like are the best descriptive statistics of the item. For example, stationeries are divided by origination, this is because Japanese and Chinese stationeries are really famous in terms of price, quality and value. Homeware is divided by rooms so that individuals can browse and select the items for the specific rooms that they are interested in. Tableware are divided by occasions, is this a fancy dinner or a normal Thursday night? People need different tableware for different occasions and it will be easier to shop on occasions. Toys are divided by age-groups so that parents and gift givers can find the items to give out more easily. Finally, accessories are divided by gender for both male and female shoppers.

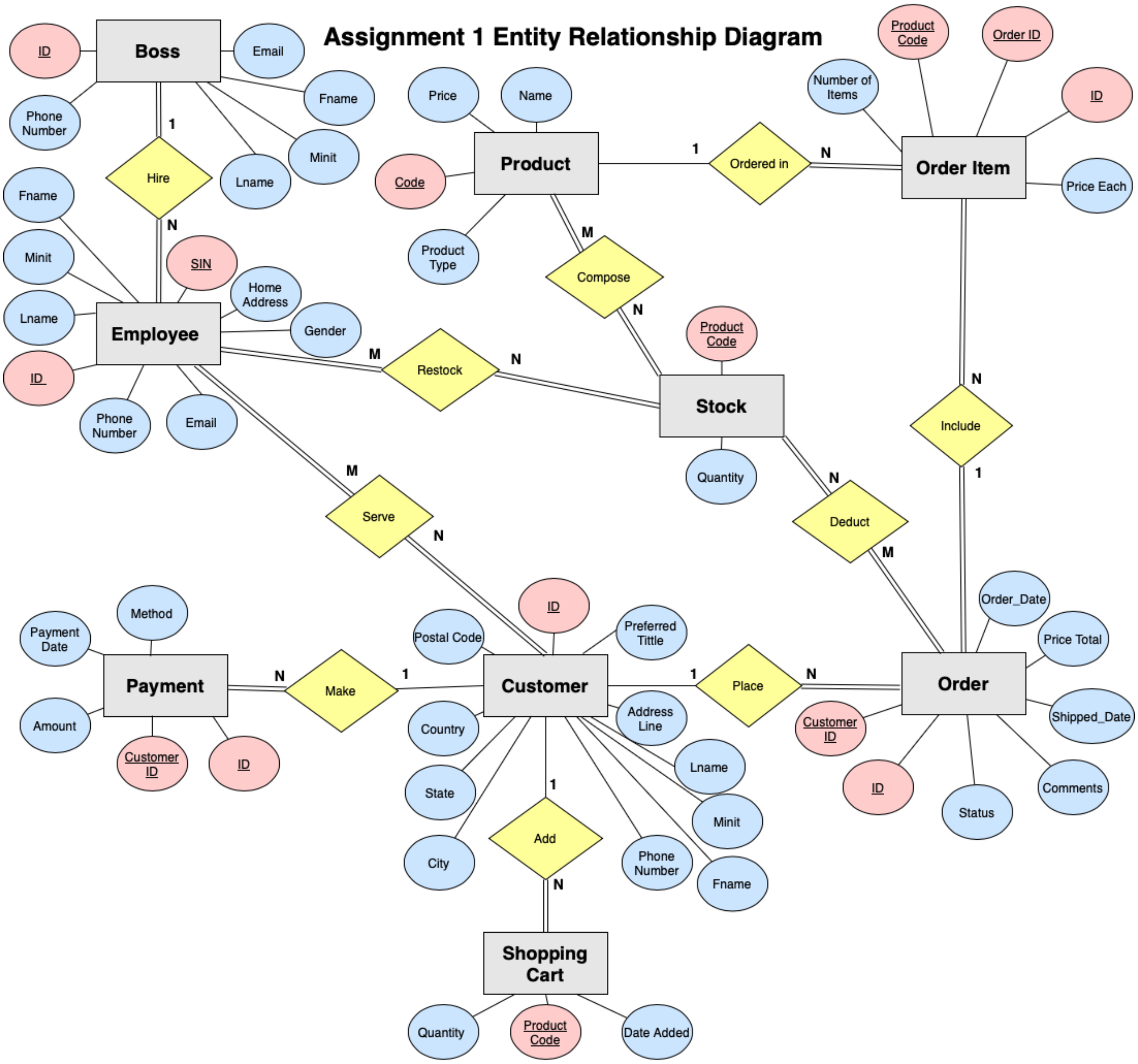
The life cycle of a customer starts by entering the website, where they will be prompted to browse items and load shopping carts. If there are any issues, customers can find employees if needed. After customers finish shopping, they will be prompted to fill in information regarding their identification. After such, they will be prompted to order and pay for the order. Order will then deduct the current stock and compose of the items that the customer has ordered. Individual ordered items included in order will be presented and received from the product region.

3. Diagram Analysis

Entities and attributes

- **Employee** (ID, SIN, Fname, Minit, Lname, Gender, Working Slot, Home Address, Phone Number, Email)
- **Boss** (ID, Fname, Minit, Lname, Phone Number, Email)
- **Customer** (ID, Fname, Minit, Lname, Preferred Title, Address Line, City, State, Country, Postal Code, Phone Number)
- **Product** (Code, Price, Name, Product Type)
- **Shopping Cart** (ID, Product Code, Quantity, Date Added)
- **Order** (ID, Order_Date, Shipped_Date, Price Total, Status, Comments, Customer ID)
- **Order Item** (ID, Order ID, Product Code, Number of items, Price Each)
- **Stock** (Product Code, Quantity)
- **Payment** (ID, Customer ID, Method, Payment Date, Amount)

3.1 ERD

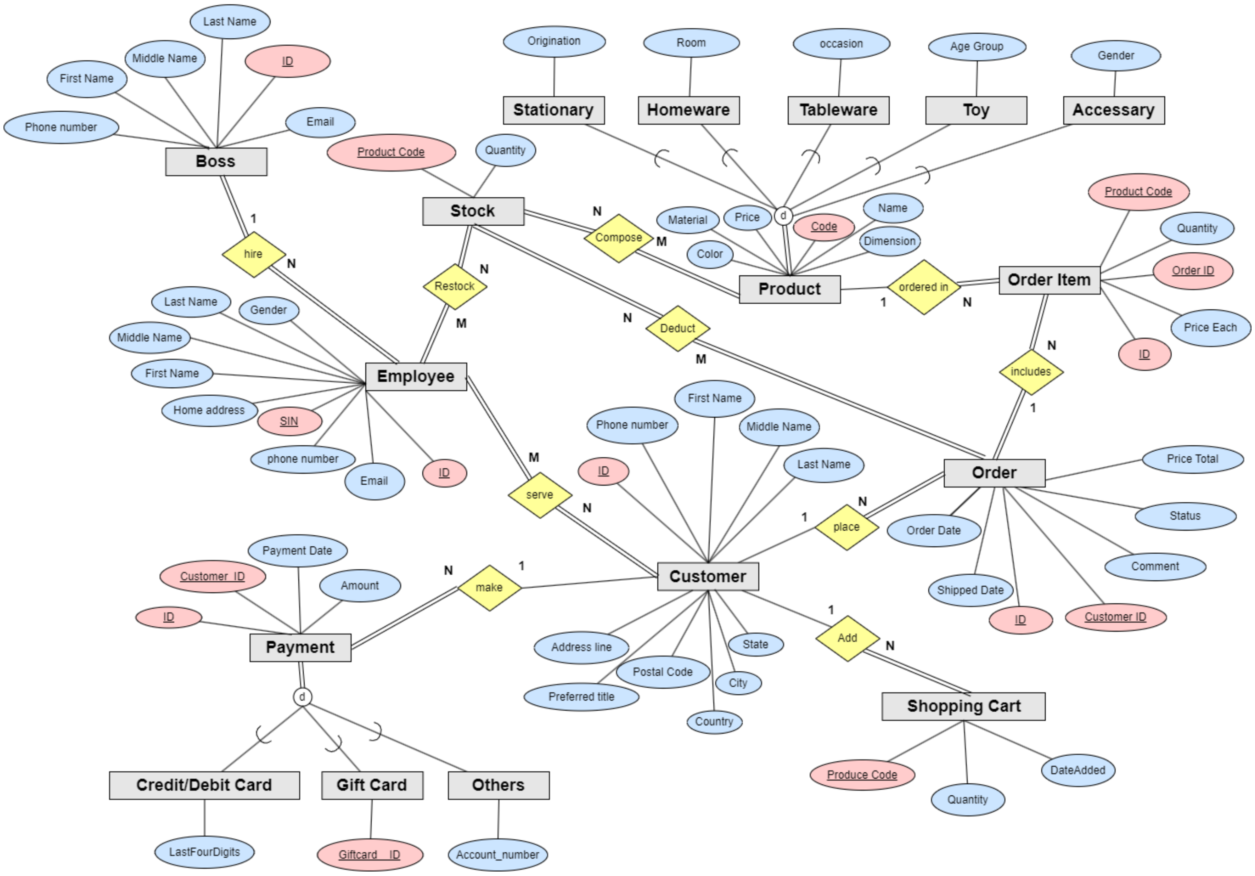


The EER diagram is a more detailed version of the ER diagram giving more information regarding the specific entities.

* For EER Diagram, there are five subclasses extended for the attribute Product Type under the entity Product (each subclass has three attributes): Stationary(Material, Color, Size) , Homeware(Material, Color, Size) , Tableware(Material, Color, Size) , Toy(Material, Color, Size) , Accessary(Material, Color, Size). Four subclasses extended for the attribute Method under the entity Payment: Credit/Debit Card, Gift Card, Others.

3.2 EERD

Assignment 1: EERD Table



Relationship Explanation

- Hire is a relationship type that relates to the entity Boss and Employee, since each boss can hire one employee, or several employees. The participation constraint on Boss's participation in this relationship type is total participation. An employee can only be hired by one boss, thus, the participation constraint on Employee's participation in this relationship type is total participation. The cardinality ratio is one to many.

- Serve is a relationship type that relates to the entity Employee and Customer, since each employee will help several customers. The participation constraint on Employee's participation in this relationship type is total participation. A customer can be served by several employees, thus, the participation constraint on Customer's participation in this total participation. The cardinality ratio is many to many.

- Restock is a relationship type that relates to the entity Employee and Stock, since each employee will restock several store stocks. The participation constraint on Employee's participation in this relationship type is total participation. The stock can be restocked by several employees, thus, the participation constraint on Stock's participation in this total participation. The cardinality ratio is many to many.

- Make is a relationship type that relates to the entity Customer and Payment, since each customer can make one payment,or several employees. Gifting and giving gift cards will create customer profiles without the payment action. The participation constraint on Customer's participation in this relationship type is partial participation. Every payment must and can only be paid by one customer, thus, the participation constraint on Payment's participation in this total participation. The cardinality ratio is one to many.

- Add is a relationship type that relates to the entity Customer and Shopping Cart, since each customer can add several shopping carts or an empty shopping cart. The participation constraint on Customer's participation in this relationship type is partial participation. Each shopping cart must and can only be added by one customer, thus, the participation constraint on Shopping Cart's participation in this total participation. The cardinality ratio is one to many.

- Place is a relationship type that relates to the entity Customer and Order, since each customer can place many orders, but not every customer needs to place an order. The participation constraint on Customer's participation in this relationship type is partial participation. An order must and can only be placed by one customer, thus, the participation constraint on Order's participation in this total participation. The cardinality ratio is one to many.

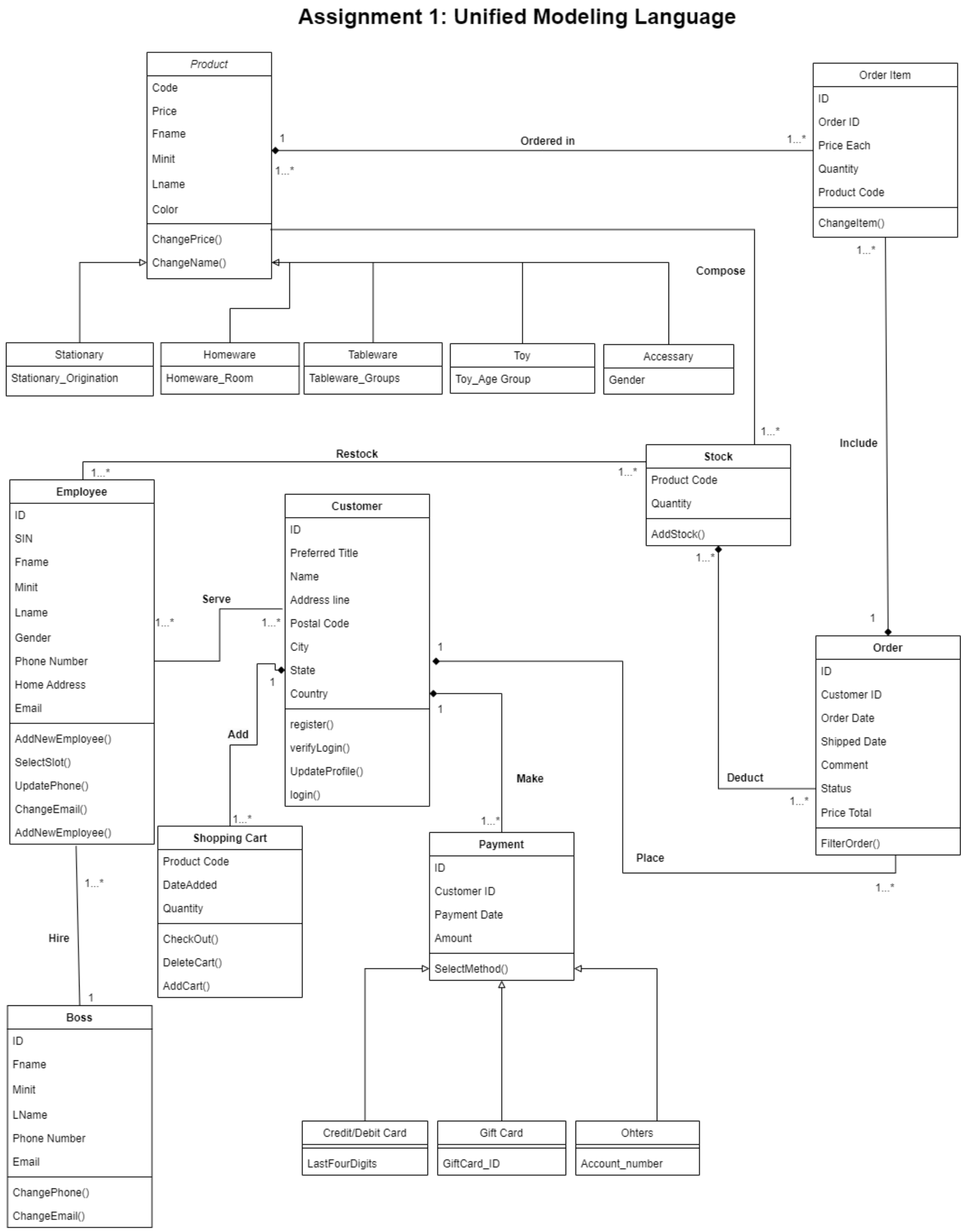
- Deduct is a relationship type that relates to the entity Order and Stock, since each order will deduct stock. The participation constraint on Order's participation in this relationship type is total participation. The stock will be deducted by placing orders, thus, the participation constraint on Stock's participation in this total participation. The cardinality ratio is many to many.

- Include is a relationship type that relates to the entity Order and Order Item, since each order will include at least one order item. The participation constraint on Order's participation in this relationship type is total participation. An order item must and can only be included in one order, thus, the participation constraint on Employee's participation in this total participation. The cardinality ratio is one to many.

- Ordered in is a relationship type that relates to the entity Order Item and Product, since every order item orders in products. The participation constraint on Order's participation in this relationship type is total participation. Not every product will be ordered in the order, thus, the participation constraint on Product's participation in this partial participation. The cardinality ratio is one to many.

- Compose is a relationship type that relates to the entity Product and Stock, since products compose stock. The participation constraint on Product's participation in this relationship type is total participation. The stock has to be composed by products, thus, the participation constraint on Stock's participation in this total participation. The cardinality ratio is many to many.

3.3 UML Diagram



UML diagram explanation
The Unified Modeling Language diagram illustrates the working process of this business with detailed attributes and applicable functions. It consists of 17 entities, including two parent and eight child entities. The relationships in the UML diagram can be classified into three categories: Inheritance, composition, and association.

- Inheritance, also known as generalization, depicts the relationship between the parent entity and the child entity. The 'Product' entity is the parent entity that is further divided into five child entities based on product types: Stationary, Tableware, Homeware, Toy, and Accessory. Since any product in this store must belong to one of the subclasses of 'Product', all child entities inherit all attributes and functions listed in the 'Product' entity, such as 'Code', 'Price', 'Name', and 'Color'. However, each child entity also has its unique attribute in addition to those listed in the 'Product' entity. For example, the 'Stationary' entity has attributes such as 'Stationary_material' and 'Stationary_size', in addition to those listed in the 'Product' entity. Similar relationships exist for the other four child entities.Another parent entity is 'Payment' with three child entities: 'Debit/credit card', 'Gift card', and 'Others'. Since the payment method of the customer must belong to any of these three options, each child entity has its unique attributes in addition to those in the 'Payment' entity. For instance, the 'Credit/debit Card' entity has an additional attribute called 'LastFourDigits', which is a variable storing the last four digits of the card number.

- Another relationship we created in the working process is called Composition relationship, in which one identified entity is a part of another entity and the smaller part cannot exist independently without the major one. Since ‘Order items’ are ordered in the ‘Product’ entity, ‘Order items’ will not exist when the ‘Product’ entity is destroyed. The multiplicity of this relationship to show the number of objects is one to one-and-more, which is what we stated in the previous section as one to many.

- The same relationship applies between ‘Order’ and ‘Order items’. Since order items are included in the order each customer placed, so ‘Order item’ does not exist if ‘Order’ entity is destroyed. When a customer places orders, the product in the stock will be deducted. But if a product is out of stock, the customer is unable to place any order. ‘Order’ cannot exist when ‘Stock’ is destroyed. Customers are people who place online orders. However, if there is no customer browsing the store website, no orders will be placed. Therefore, ‘Order’ cannot exist when the ‘Customer’ entity is destroyed. Last but not the least, customers make payments after they place orders. But ‘payment’ cannot exist independently without ‘Customer’.

4. Diagram Comparison

In this assignment, we are tasked to make comparisons between three different diagrams, the ERD, EERD and UML. These are all modelling techniques that are used to show the database structure of a given mini-environment. After creating a diagram using each of the techniques, we have been able to make comparisons between the strength and weakness of each of the models.

Let us first discuss the strength of the ERD model. The ERD models fulfill the basic requirements of the database creation. It creates a simplistic and direct visualization of the database. It is simple and easy to understand even with the least amount of technical background. The ERD diagram clearly presents different entities and attributes along with the relationship between them.

However, there are still some weaknesses demonstrated by the ERD diagram. For example, it cannot show all the required and detailed features in the database. Furthermore, it cannot portray complex and a higher amount of relationships between entities and attributes, which could be accomplished in EERD and UML diagrams. Lastly, it can not show the complete description of the product type, unlike EERD and UML diagrams.

Moving on to the EERD. We observe all the strengths that the ERD diagram presents such as visualization, easiness to understand and the ability to portray relationships. In addition to ERD diagrams, there are some extended features such as the abilities to create subclasses & superclasses, the choice of converging generalization or diverging into specializations, creating categories and allowing relationship inheritance between entities. Subclasses & superclasses relationships, it can allow the demonstration of different product types, payment methods and more.

There are a couple weaknesses for EERD in comparison to the other two methods. We first see that it is a little more difficult to understand EERD without knowing about specialization & generalization or subclasses & superclasses. Additionally it gets really messy once we input a hefty amount of data so it cannot demonstrate a large amount of detailed database design well.

Finally, we will examine the UML model. The UML model is the most complex database modelling system among the three listed above. It is able to model complex data and interactions and store a mass amount of entities and relationships. It is perfect for managing complex data with complex relationships. It could also store information by type, which is unseen in the other two models.

However, the UML model is not easy to understand and requires a big amount of effort in terms of teaching and educating the reader into reading the UML model to those who are not familiar with the UML model.

In terms of aesthetics, we observe that ERD & EERD models have the ability to become colour coded and demonstrate the required information. In contrast, UML diagrams are a lot more dull in presentation and more business format.

Combining the strengths and weaknesses from the above diagrams, we will be able to present different diagrams using different models for presentations in different occasions and in different lifespans of a project. For example, ERD diagrams will be very good for preliminary thinking during the early stages of a project. EERDs will be great after receiving feedback and gaining traction. UMLs will be for formal business presentations in which we need to demonstrate the organization towards skilled professionals. If the stakeholder is not as skilled, I would actually argue that EERD has strength over UML. This is because we would have needed to spend precious presentation time explaining how to read UML diagrams.

For internal and team use cases. EERD can be more agile in manipulation of different entities and describe the relationship between them well. However, UML diagrams have a better proficiency by including the storage data type. This is why I would prefer UML diagrams over EERD in the specific scenario. Since there would be a level of anticipation that individuals working would be rather familiar with the information included and it would not take too long for those who do not have a basic understanding of the model to get started to learn about it.

5. Data Catalog

Attributes	**Entity**	**Type**	**Domain**	**Constrain**	**Key**
Code	**Product**	STRING	CHAR(9) 9 character combination of letters and number	- Not NULL - Unique	Primary Key
Price		FLOAT	The set of floats larger than 0.00	- Not NULL - Currency: Canadian dollars (CAD)	
Fname		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination	- Not NULL	
Minit		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination	- Not NULL	
Lname		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination	- Not NULL	
Color		STRING	VARCHAR(100) Maximum of 20 characters combination (eg. Red spots with yellow lines)	- Not NULL	
STOCK_ORDER_ID		INTEGER	INT(4) (eg. 1401)	-Not NULL -Unique	Foreign key
ID	**Order Item**	STRING	CHAR(5) 5 of letters and	- Not NULL - Unique	Primary key

			numbers		
Price Each		FLOAT	The set of floats larger than 0.00	- Not NULL - Currency: Canadian dollars (CAD)	
Quantity		INTEGER	Integers larger than 0	- Not NULL	
Product Code		STRING	CHAR(9) 9 character combination of letters and number	- Not NULL - Unique	Foreign key
ORDER_ID		INTEGER	INT()	- Not NULL - Unique	Foreign Key
Product_Cod e		STRING	CHAR(9)	- Not NULL - Unique	Foreign Key
ID	**Order**	INTEGER	INT(4) (eg. 1401)	- Not NULL - Unique	Primary key
Customer ID		STRING	CHAR(12) 12 character combination of letters and number; first two characters are letters	- Not NULL - Unique	Foreign key
Order Date		DATE	MM-DD-YYYY (eg. 03-06-2023)	- Not NULL	
Shipped Date		DATE	MM-DD-YYYY (eg. 03-06-2023)	- Not NULL	
Comment		STRING	VARCHAR(800) (eg. please handle the package to my concierge)		
Status		ENUMERATED	ENUM	- Not NULL	

			(‘Confirmed’ , ‘Processing’, ‘Shipping, ‘Delivered’, ‘Canceled’)		
Total Price		FLOAT	The set of floats larger than 0.00	- Not NULL - Currency: Canadian dollars (CAD)	
Product Code	**Stock**	STRING	CHAR(9) 9 character combination of letters and number	- Not NULL - Unique	Foreign key
Quantity		INTEGER	Integers larger than 0	- Not NULL	
ID	**Employee**	STRING	CHAR(7) 7 character combination of letters and numbers; first two characters are numbers	- Not NULL - Unique	Primary key
SIN		STRING	CHAR(9) A series of 9 integers	- Not NULL - Unique	
Fname		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination	- Not NULL	
Minit		STRING	VARCHAR(45) Maximum of 45 alphabetical letters		

			combination		
Lname		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination	- Not NULL	
Gender		ENUMERATED	ENUM(‘Male’, ‘Female’, ‘Prefer not to say’)	- Not NULL	
Phone Number		STRING	CHAR(10) A series of 10 integers	- Not NULL	
Home Address		STRING	VARCHAR(100) Maximum of 100 alphabetical letters and numbers combination	- Not NULL	
Email		STRING	VARCHAR(50) A series of characters with length smaller than 50 letters	- Not NULL	
BOSS_ID		STRING	CHAR(7)	- Not NULL - Unique	Foreign key
ID	**Customer**	STRING	CHAR(12) 12 characters combination of letters and numbers; first two characters are letters	- Not NULL - Unique	Primary key
Preferred Title		ENUMERATED	ENUM(‘Mr’, ‘Mrs’, ‘Miss’. ‘Ms’)	- Not NULL	
Fname		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination	- Not NULL	

Minit		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination		
Lname		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination	- Not NULL	
Address Line		STRING	VARCHAR(100) Maximum of 100 alphabetical letters and numbers combination	- Not NULL	
Postal Code		STRING	CHAR(6) A combination of letters and numbers	- Not NULL	
City		STRING	VARCHAR(100) Maximum of 100 alphabetical letters combination	- Not NULL	
State		STRING	VARCHAR(50) Maximum of 50 alphabetical letters combination	- Not NULL	
Country		STRING	VARCHAR(50) Maximum of 50 alphabetical letters combination	- Not NULL	
Employee and Customer: Many to many FK					

Table					
CUSTOMER_ ID		STRING	CHAR(12)	- Not NULL - Unique	Foreign key
EMPLOYEE_ ID		STRING	CHAR(7)	- Not NULL - Unique	Foreign key
Product Code	**Shopping Cart**	STRING	CHAR(9) 9 character combination of letters and number	- Not NULL - Unique	Foreign key
Date Added		DATE	MM-DD-YYYY (eg. 03-06-2023)	- Not NULL	
Quantity		INTEGER	Integers larger than 0	- Not NULL	
CUSTOMER_ ID		STRING	CHAR(12)	- Not NULL - Unique	Foreign key
ID	**Payment**	STRING	CHAR(10) 10 character combination of letters and numbers	- Not NULL - Unique	Primary key
Payment Date		DATETIME	MM-DD-YYYY HH:MI:SS (eg. 03-06-2023 13:21:09)	- Not NULL	
Amount		FLOAT	Float larger than 0.00	- Not NULL	
CUSTOMER_ ID		STRING	CHAR(12)	- Not NULL - Unique	Foreign key
ID	**Boss**	STRING	CHAR(7) 7 character combination of letters and numbers; first two characters are numbers	- Not NULL - Unique	Primary key

Fname		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination	- Not NULL	
Minit		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination		
Lname		STRING	VARCHAR(45) Maximum of 45 alphabetical letters combination	- Not NULL	
Phone Number		STRING	CHAR(10) A series of 10 integers	- Not NULL	
Email		STRING	VARCHAR(50) A series of characters with length smaller than 50 letters	- Not NULL	

6. Reference

Due to the need for privacy, we are unable to provide the citation of the online store that we have interviewed.

7. Statement of Individual Contributions

Executive Summary: Kuan Yi Chou
Context for the Study: Kuan Yi Chou
Modelling using ERD: YiRan Zheng
Modelling using EERD: Ruolan Zhang
Modelling using UML diagram: Ruolan Zhang & YiRan Zheng
Comparison of the Three Modeling Techniques: Kuan Yi Chou
Data catalog: Kuan Yi Chou & Ruolan Zhang & YiRan Zheng
References (1 page, APA format)
Ideas were generated between all three individuals.