

TABLE OF CONTENTS

- Summary of Company Overview and Database Planning
- Revised Database Design
 - Logical ERD Modeling
 - Normalization
- Physical Database Design
- User Interface Design

UNIQLO COMPANY PROFILE

PT. Fast Retailing Indonesia (UNIQLO) is a clothing retail company based in South Jakarta, established on October 16, 2012, with 40 stores across Indonesia (as of March 31, 2021). UNIQLO, originally established in Yamaguchi, Japan in 1949 as a textiles manufacturer, has grown into a global apparel brand with more than 1,000 stores worldwide. The company has maintained its commitment to redefining clothing, emphasizing quality and textiles since its inception.



DATABASE PLANNING

Mission Statement:

To create a robust and scalable database model that supports Uniqlo's offline retail business process, enabling seamless data accessibility, security, and performance for a superior customer experience and business operations.

Mission Objective:

- To maintain (enter, update, and delete) data on customers, orders, coupons, staffs, payments, order_details, products.
- To perform searches on customers, orders, coupons, staffs, payments, order_details, products.
- To report on customers, orders, coupons, staffs, payments, order_details, products.

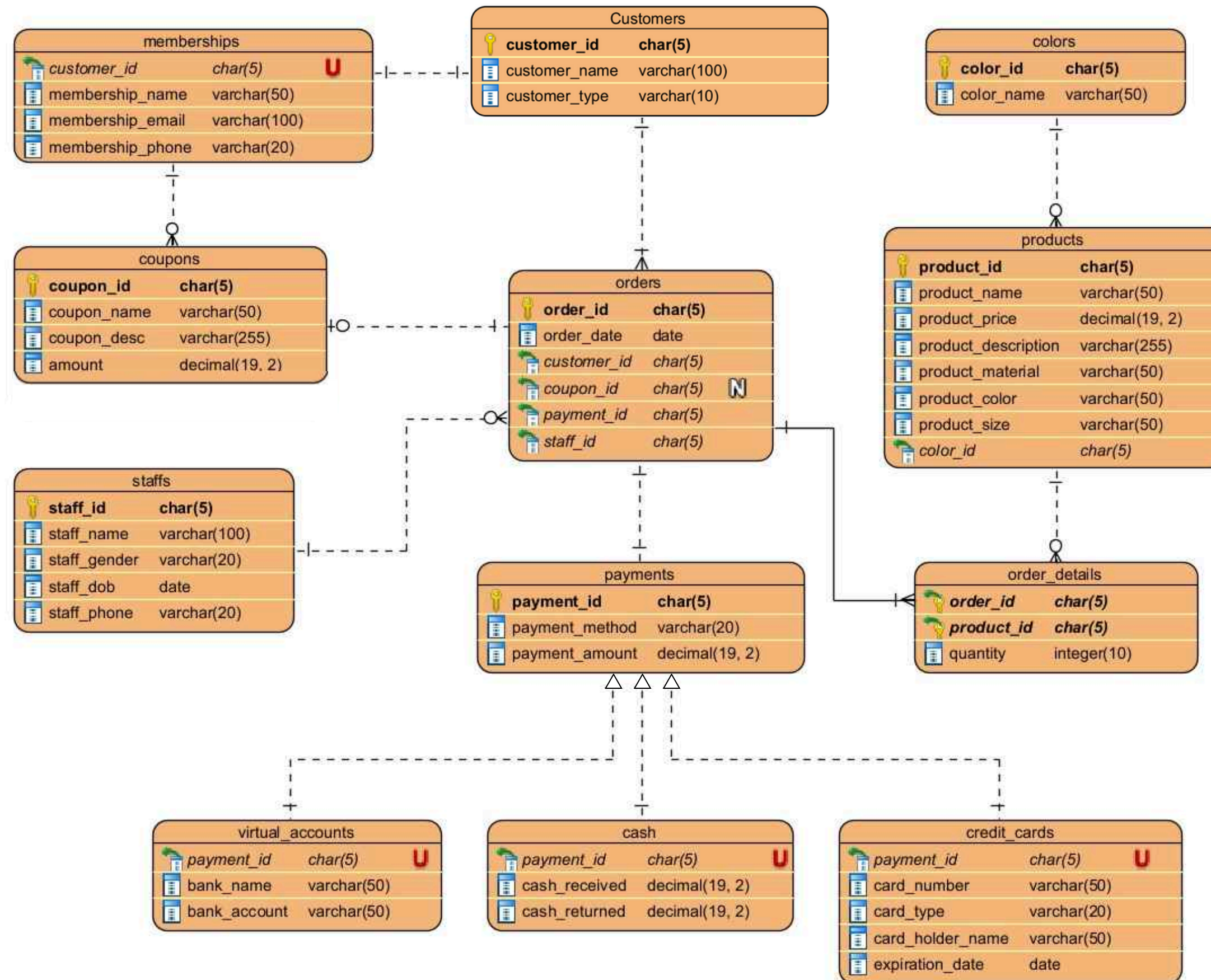


Database Design

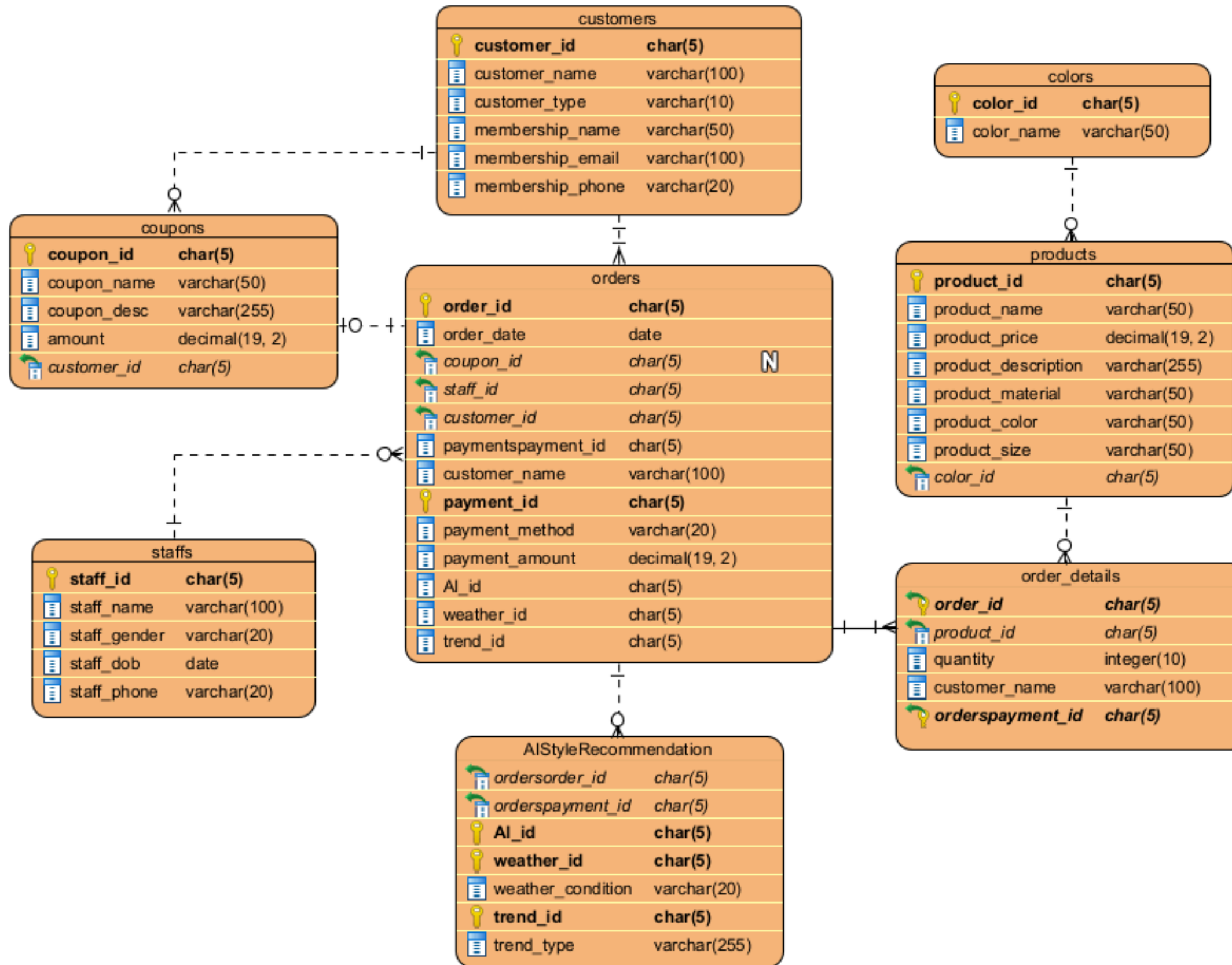
NORMALIZATION

Step 1.1				
Entity Name	Description	Alliases	Occurence	
Customers	Represent a customer who buys a product or being a member	Customer	Each customer that join order a product or memberships program using app.	
Products	Represents the items that are available for sale.	Items, Goods, Merchandise	Discussed in relation to order and product details.	
Staffs	General term for staff employed by Uniqlo	Employee	Staff works and handle orders.	
Orders	Item(s) purchased by a customer	Order	Linked to customers during every transactions.	
Coupons	Coupon thats only belongs to customers who is membership	Voucher	Each member may have or use their coupon for orders.	
Payments	Represents a payment associated with customers	Receipt, Billing	Associated transaction with customer.	
Step 1.2				
Conceptual Design Step	Normalisasi			» »

LOGICAL DATABASE DESIGN



DENORMALIZATION



PHYSICAL DATABASE DESIGN

STEP 3.1 Design Base Relations

DBDL for Customers

Domain CustomerID:	variable length character string, length 5
Domain CustomerType:	variable length character string, length 20

```
Customers(  
    customer_id      SupplierID    NOT NULL,  
    customer_type    CustomerType  NOT NULL,  
    PRIMARY KEY (customer_id) );
```


DBDL for Memberships

Domain MemberName:	variable length character string, length 5
Domain MemberPhone:	variable length character string, length 12
Domain MemberDOB:	DATE
Domain MemberEmail:	variable length character string, length 50

Memberships(

member_name	MemberName	NOT NULL,
member_phone	MemberPhone	NOT NULL,
member_dob	MemberDOB	NOT NULL,
member_email	MemberEmail	NOT NULL,

PRIMARY KEY (customer_id)

FOREIGN KEY (customer_id) REFERENCES Customers(customer_id) ON UPDATE CASCADE
ON DELETE CASCADE);

DBDL for Customers & Memberships

```
CREATE TABLE Customers(  
customer_id CHAR(5) PRIMARY KEY,  
customer_type VARCHAR(100) NOT NULL,  
);
```

```
CREATE TABLE Memberships(  
customer_id CHAR(5) PRIMARY KEY,  
member_name VARCHAR(100) NOT NULL,  
member_phone VARCHAR(100) NOT NULL,  
member_dob DATE,  
FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id)  
);
```


DBDL for Coupons

Domain CouponID:	variable length character string, length 5
Domain CouponName:	variable length character string, length 100
Domain CouponDescription:	variable length character string, length 100
Domain CouponAmount:	integer, in the range 1-15

Coupons(
coupon_id	CouponID	NOT NULL,
coupon_name	CouponName	NOT NULL,
coupon_description	CouponDescription	NOT NULL,
amount	CouponAmount	NOT NULL,
PRIMARY KEY (coupon_id));		

DBDL for Orders

Domain OrderID:	variable length character string, length 5
Domain OrderDate:	DATE

Orders(
order_id	OrderID	NOT NULL,
order_date	OrderDate	NOT NULL,
customer_id	CustomerID	NOT NULL,
coupon_id	CouponID	NOT NULL,
payment_id	PaymentID	NOT NULL,
staff_id	StaffID	NOT NULL,

PRIMARY KEY (coupon_id)
FOREIGN KEY (customer_id) REFERENCES Customers(customer_id) ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (coupon_id) REFERENCES Coupons(coupon_id) ON UPDATE CASCADE ON DELETE CASCADE),
FOREIGN KEY (payment_id) REFERENCES Payment(payment_id) ON UPDATE CASCADE ON DELETE CASCADE),
FOREIGN KEY (staff_id) REFERENCES Staffs(staff_id) ON UPDATE CASCADE ON DELETE CASCADE));

DBDL for Coupons and Orders

```
CREATE TABLE Coupons(  
coupon_id VARCHAR(5) PRIMARY KEY,  
coupon_name VARCHAR(100) NOT NULL,  
coupon_description VARCHAR(100) NOT NULL,  
amount int NOT NULL  
);
```

```
CREATE TABLE Orders(  
order_id CHAR(5) PRIMARY KEY,  
order_date DATE NOT NULL,  
);
```

DBDL for Orders_details

```
CREATE TABLE Order_details (  
    order_id CHAR(5) NOT NULL,  
    product_id CHAR(5) NOT NULL,  
    quantity INT NOT NULL,  
    PRIMARY KEY (order_id, product_id),  
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),  
    FOREIGN KEY (product_id) REFERENCES Products(product_id)  
);
```


DBDL for Products

```
CREATE TABLE Products (  
    product_id CHAR(5) PRIMARY KEY NOT NULL,  
    product_name VARCHAR(50) NOT NULL,  
    product_price DECIMAL(19, 2) NOT NULL,  
    product_description VARCHAR(255) NOT NULL,  
    product_material VARCHAR(50) NOT NULL,  
    product_color VARCHAR(50) NOT NULL,  
    product_size VARCHAR(50) NOT NULL,  
);
```

DBDL for Colors

```
CREATE TABLE Colors (  
    color_id CHAR(5) PRIMARY KEY NOT NULL,  
    color_name VARCHAR(100) NOT NULL  
);
```

DBDL for Staffs

```
CREATE TABLE Staffs (  
    staff_id CHAR(5) PRIMARY KEY,  
    staff_name VARCHAR(100) NOT NULL,  
    staff_gender VARCHAR(20) NOT NULL,  
    staff_dob DATE NOT NULL,  
    staff_phone VARCHAR(20) NOT NULL  
);
```


DBDL for Payments

```
CREATE TABLE Payments (  
    payment_id CHAR(5) PRIMARY KEY,  
    payment_method VARCHAR(20) NOT NULL,  
    payment_amount DECIMAL(19,2) NOT NULL  
);
```

DBDL for VirtualAccount

```
CREATE TABLE Virtual_accounts (  
    payment_id CHAR(5) NOT NULL UNIQUE,  
    bank_name VARCHAR(100) NOT NULL,  
    bank_account VARCHAR(100) NOT NULL,  
    FOREIGN KEY (payment_id) REFERENCES  
    Payments(payment_id)  
);
```

DBDL for Cash

```
CREATE TABLE Cash (  
    payment_id CHAR(5) NOT NULL UNIQUE,  
    cash_received NOT NULL,  
    cash_given DECIMAL(19,2) NOT NULL,  
    FOREIGN KEY (payment_id) REFERENCES  
    Payments(payment_id)  
);
```

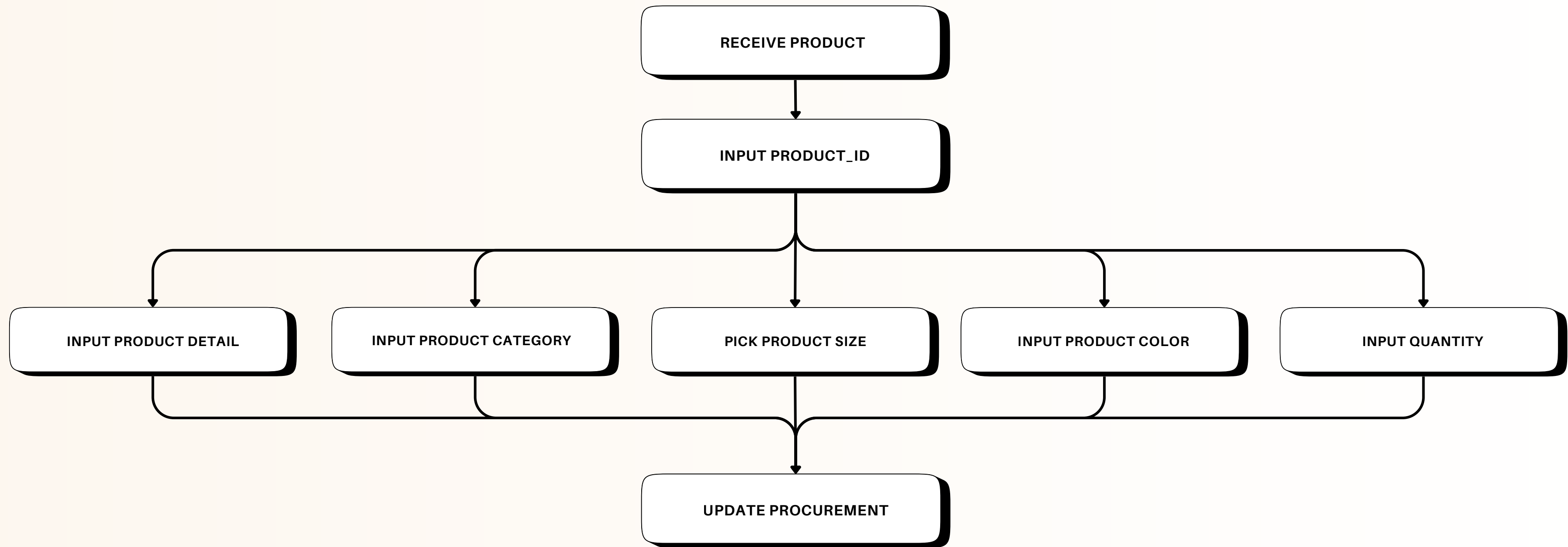
DBDL for CreditCard

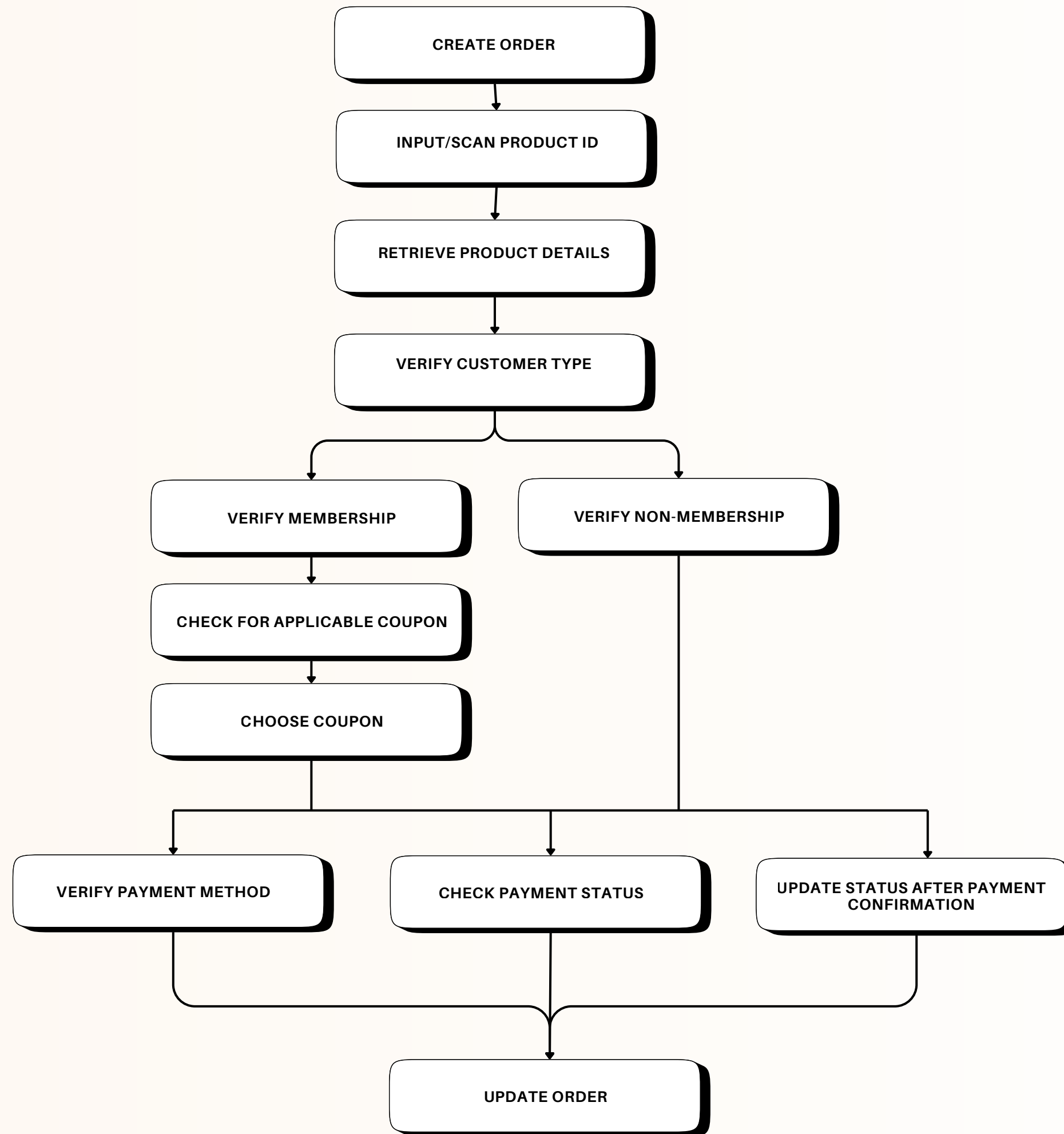
```
CREATE TABLE Credit_card (  
    payment_id CHAR(5) NOT NULL UNIQUE,  
    card_number VARCHAR(50) NOT NULL,  
    card_type VARCHAR(20) NOT NULL,  
    card_holder_name VARCHAR(100) NOT NULL,  
    expiration_date DATE NOT NULL,  
    FOREIGN KEY (payment_id) REFERENCES Payments(payment_id)  
);
```


DBDL for AIStyleRecommendation

```
CREATE TABLE AIStyleReccomendation (  
    order_id CHAR(5) NOT NULL,  
    payment_id CHAR(5) NOT NULL,  
    AI_id CHAR(5) NOT NULL,  
    weather_id CHAR(5) NOT NULL,  
    weather_condition VARCHAR(20) NOT NULL,  
    trend_id CHAR (5) NOT NULL,  
    trend_type VARCHAR(255) NOT NULL,  
    PRIMARY KEY(AI_id, weather_id, trend_id),  
    FOREIGN KEY (payment_id) REFERENCES Orders(payment_id),  
    FOREIGN KEY (order_id) REFERENCES Orders(order_id)  
);
```

TRANSACTION DESIGN





AI implementation on Uniqlo Offline POS

Real-Time Demand Prediction:

AI Predict the demand for specific products at each store location by analyzing:

- Foot traffic patterns using **in-store sensors** or **cameras**.
- Historical sales trends
- External factors like local events, holidays, and weather conditions.

Shelf Heatmaps and Replenishment Alerts:

Integrate with **in-store sensors** or RFID technology to monitor which shelves or sections attract the most attention.

- If certain items on a shelf are frequently picked up but not purchased, **AI** can detect and recommend better placement or promotions.

Additional Attribute:

- Weathers
- Trend_Search

User Interface Design

Cashier Dashboard

Manage Products

Manage Orders

Manage Products

Product ID

Product Name

Select Size

Color

Select Category

Stock

Price

Add Product

#	ID	Name	Size	Color	Category	Stock	Price	Actions
---	----	------	------	-------	----------	-------	-------	---------

<https://kennedysurianto.github.io/dim-uniqlo/>

User Interface Design (AI Implementation)

 Tokyo, Japan



December

2024

 20°C

Most Searched Queries(Updated Real-time):

1

Sweater

2

Hoodie

3

Baju Polos Putih





4

Jaket Anti UV

5

Baju Airism

Style Recommendation:





Style 1

Product Recommendation:



1. Sweater Blue



2. Heatech Syal



3. Kaos UT Blue



4. Celana Jogger

THANK YOU!