



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

DATABASE

SECD2523

SECTION 10

LECTURER: DR. ROZILAWATI

GROUP NAME: TECHTITANS

No	Name	Matric No.
1	Puventhiran A/L Ganeson	A22EC0262
2	Ee Ze Qing	A22EC0155
3	Winki Wong Jia Xuan	A22EC0116
4	Mok Yun Yi	A22EC0078

Video link: [Database_project_video_techtitans.mp4](#)

Table of contents

1.0 Introduction.....	3
2.0 Overview of Project.....	4
3.0 Database Conceptual Design.....	4
3.1 Updated business rule.....	4
3.2 Conceptual ERD.....	5
4.0 Database Logical Design.....	6
4.1 Logical ERD.....	6
4.2 Updated Data Dictionary.....	7
4.2.1 Description of Entity.....	7
4.2.2 Description of Relationship.....	8
4.2.3 Description of Attributes.....	9
4.3 Normalisation.....	11
5.0 Relational DB Schemas (after normalisation).....	14
6.0 SQL Statements (DDL & DML).....	16
7.0 Summary.....	32

1.0 Introduction

Nowadays, there has been a significant shift in the e-commerce market that has changed how companies and consumers place their orders. The rise of digital commerce platforms was not only easier for the shopping experience but also revolutionised how sellers and retailers interact with their clients. Our e-commerce system, rightfully called “Vrify”, takes precedence from the groundbreaking success of “Shopee”. This innovative e-commerce platform raised standards in the Asian region and beyond, within this dynamic and always-changing environment.

In appreciation of Shopee’s amazing impact and outstanding accomplishments in the field of online shopping, the name of our e-commerce system “Vrify” pays respect to the company. With Vrify, we are hoping to expand the insightful lessons we’ve learned from Shopee’s experience while also offering our own viewpoint and proactive solutions to meet the specific needs and tastes of our target market.

We have far bigger plans for Vrify than just the typical. After giving careful consideration to the difficulties encountered by today’s e-commerce customers, we aim to include features that influence the direction of e-commerce in the future. Our ongoing commitment to quality drives us to deliver a personalised, fast, secure purchasing experience that caters to your individual tastes regardless of the location of the desired item. We have bigger plans for Vrify than just conventional e-commerce. We’ve made use of cutting-edge technology to provide customers with an unconventional shopping experience.

Through the use of augmented reality (AR) and virtual reality (VR) elements, shoppers may go on a shopping trip that includes not just transactions but also experiences and items. Before making a purchase, have the ability to digitally try on apparel, accessories, or even house furniture. Vrify’s augmented capabilities let you see items in your real environment, guaranteeing that you will receive exactly what you see. Whether you are remodelling your living space or your wardrobe, Vrify’s AR integration will help everybody to make more accurate and precise decisions.

2.0 Overview of Project

After completing the second phase, we have identified and outlined the problem statements and conceptualised the database for our e-commerce system, "Vrify." As we enter the database design phase, we aim to enhance the system by updating the conceptual design, which includes updating the business rules and Conceptual Entity-Relationship Diagram (ERD). The next steps followed by translating the conceptual ERD into a logical ERD, creating relation schemas, and normalising data to enhance logical storage efficiency.

Then, we will update the data dictionary and normalised relations. The process then will proceed to implement the logical database design and generate SQL statements for the system's database. This approach ensures that the database seamlessly aligns with our goal of refining the e-commerce experience on the "Vrify" platform.

3.0 Database Conceptual Design

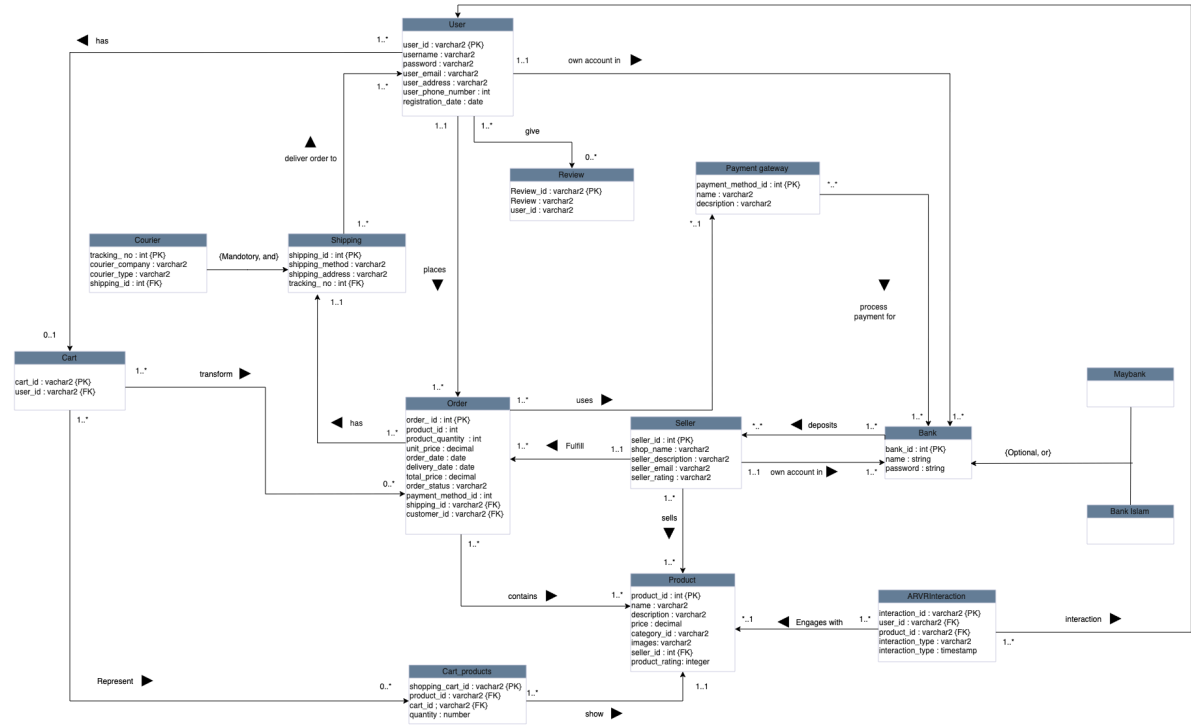
3.1 Updated business rule

1. The system operates 7 days a week and 24 hours per day.
2. Customer accounts need to be created with a valid email address and phone number and personal information.
3. Sellers need to register an account with a real business registration name with SSM.
4. Sellers are required to verify business licences before registering an account.
5. Customers are able to access AR and VR interaction when shopping for a product.
6. Customers can place orders through selecting wanted products that are added in the cart.
7. Orders can only be processed after successful payment authorization.
8. Payment status of the orders will be updated in 2 minutes.
9. Acceptable payment methods included credit or debit cards and online banking transfer.
10. All payment transactions must comply with PCI DSS standards.
11. Shipping options and its costs need to be stated before checkout.
12. Real-time tracking orders information is provided to customers after payment successfully.

13. When the order is received, customers are prompted to give a review of the products.

3.2 Conceptual ERD

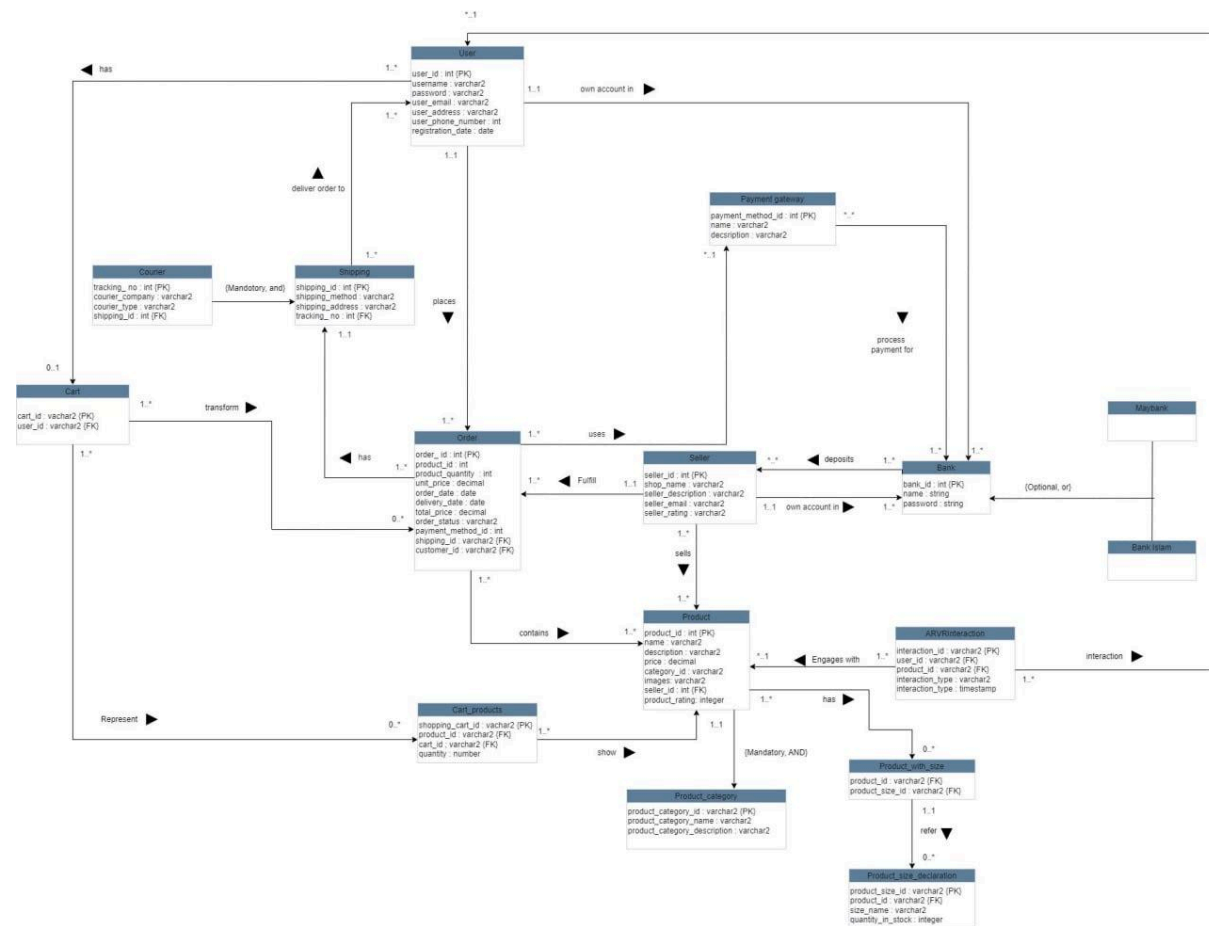
Conceptual ERD



4.0 Database Logical Design

4.1 Logical ERD

LOGICAL DATA MODEL



4.2 Updated Data Dictionary

4.2.1 Description of Entity

Entity	Description	Occurrence
USER	Hold the data of Users in the E-commerce platform.	Users register and login the account, filling their personal information.
SELLER	Hold the data of Sellers in the E-commerce platform.	Sellers register their store in the E-commerce platform, setting up store name and description.
BANK	Hold the data of Banks.	There are different banks to process the payment.
MAYBANK	Hold the data of Maybank.	The payment of the customer is easily handled by the bank.
BANK_ISLAM	Hold the data of Bank Islam.	The payment of the customer is easily handled by the bank.
ORDER	Hold the data of the Order.	An order made by the user and received by the Shipping and Payment Gateway. In the Order, the quantity, unit price, order date, total price, payment method and shipping status will be recorded.
PAYMENT_GATEWAY	Hold the data of the Payment methods and gateways.	The Payment method details such as name and description will be stored.
SHIPPING	Hold the data of Shipping.	The users' shipping details such as shipping address, shipping method and tracking number will be stored.
COURIER	Hold the data of the Courier.	Courier sends the order to customers and updates the status by tracking numbers to the user.

4.2.2 Description of Relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity
USER	1...1	Own account in	1...*	BANK
USER	1...*	Has	0...1	CART
USER	1...1	Places	1...*	ORDER
CART	1...*	Transforms	0...*	ORDER
CART	1...*	Represents	0...*	CART_products
SHIPPING	1...*	Deliver order to	1...*	USER
ORDER	1...*	Has	1...1	SHIPPING
ORDER	1...*	Contains	1...*	PRODUCT
ORDER	1...*	Uses	*...1	PAYMENT_GATEWAY
SELLER	1...1	Fulfil	1...*	ORDER
SELLER	1...1	Owens account in	1...*	BANK
SELLER	1...*	Sells	1...*	PRODUCT
BANK	1...*	Deposits	*...*	SELLER
PRODUCT	1...*	Has	0...*	PRODUCT_WITH_SIZE
ARVR_INTERACTION	1...*	Interaction to	*...1	USER
ARVR_INTERACTION	1...*	Engages with	*...1	PRODUCT
PRODUCT_WITH_SIZE	0...*	Refers	0...*	PRODUCT_SIZE_DECLARATION
PAYMENT_GATEWAY	*...*	Process payment for	1...*	BANK

4.2.3 Description of Attributes

Entity	Attribute	Description	Data Type	Constraint
USER	user_id	User's ID	VARCHAR2(20)	PRIMARY KEY
	username	User's name	VARCHAR2(20)	NOT NULL
	password	User's password	VARCHAR2(30)	NOT NULL
	user_email	User's email	VARCHAR2(30)	NOT NULL
	user_address	User's address	VARCHAR2(200)	NOT NULL
	user_phone_number	User's phone number	NUMBER(20)	NOT NULL
	registration_date	Date of user registration	DATE	NOT NULL
CART	cart_id	Cart's ID	VARCHAR2(20)	PRIMARY KEY
	user_id	User's ID	VARCHAR2(20)	FOREIGN KEY
SHIPPING	shipping_id	Shipping's ID	VARCHAR2(20)	PRIMARY KEY
	shipping_method	Method of shipping	VARCHAR2(20)	NOT NULL
	shipping_address	Address to ship	VARCHAR2(200)	NOT NULL
	tracking_no	Tracking number	NUMBER(20)	FOREIGN KEY
ORDER	order_id	Order's ID	VARCHAR2(20)	PRIMARY KEY
	product_id	Product's ID	VARCHAR2(20)	NOT NULL
	product_quality	Quality of product	NUMBER(20)	NOT NULL
	unit_price	Price per unit	DECIMAL(10,2)	NOT NULL
	order_date	Date of order placed	DATE	NOT NULL
	delivery_date	Date of delivery	DATE	NOT NULL
	total_price	Total price	DECIMAL(20,2)	NOT NULL
	order_status	Status of order	VARCHAR2(200)	NOT NULL
	payment_method_id	Payment method's ID	NUMBER(20)	NOT NULL

	shipping_id	Shipping's ID	VARCHAR2(20)	FOREIGN KEY
	customer_id	Customer's ID	VARCHAR2(20)	FOREIGN KEY
SELLER	seller_id	Seller's ID	VARCHAR2(20)	PRIMARY KEY
	shop_name	Name of shop	VARCHAR2(50)	NOT NULL
	seller_description	Description of seller	VARCHAR2(200)	NOT NULL
	seller_email	Seller's email	VARCHAR2(30)	NOT NULL
	seller_rating	Rating of seller	VARCHAR2(30)	NOT NULL
BANK	bank_id	Bank's ID	VARCHAR2(20)	PRIMARY KEY
	name	Name of bank	VARCHAR2(50)	NOT NULL
	password	Bank's password	VARCHAR2(30)	NOT NULL
PRODUCT	product_id	Product's ID	VARCHAR2(20)	PRIMARY KEY
	name	Name of product	VARCHAR2(30)	NOT NULL
	description	Description of product	VARCHAR2(200)	NOT NULL
	price	Price of product	DECIMAL(10,2)	NOT NULL
	category_id	Category's ID	VARCHAR2(20)	NOT NULL
	images	Image of product	VARCHAR2(30)	NOT NULL
	seller_id	Seller's ID	VARCHAR2(20)	FOREIGN KEY
	product_rating	Rating of product	NUMBER(20)	NOT NULL
ARVR_INTERACTION	interaction_id	Interaction's ID	VARCHAR2(30)	PRIMARY KEY
	user_id	User's ID	VARCHAR2(20)	FOREIGN KEY
	product_id	Product's ID	VARCHAR2(20)	FOREIGN KEY
	interaction_type	Type of interaction	VARCHAR2(30)	NOT NULL
	interaction_type	Type of interaction	TIMESTAMP	NOT NULL
PRODUCT_CATEGORY	product_category_id	Product categor's ID	VARCHAR2(30)	PRIMARY KEY
	product_category_name	Name of product category	VARCHAR2(30)	NOT NULL

	product_category_description	Description of product category	VARCHAR2(200)	NOT NULL
PRODUCT_WITH_SIZE	product_id	Product's ID	VARCHAR2(20)	FOREIGN KEY
	product_size_id	Product size's ID	VARCHAR2(20)	FOREIGN KEY
PRODUCT_SIZE_DESCRIPTION	product_size_id	Product size's ID	VARCHAR2(20)	PRIMARY KEY
	product_id	Product's ID	VARCHAR2(20)	FOREIGN KEY
	size_name	Name of size	VARCHAR2(30)	NOT NULL
	quantity_in_stock	Quantity in stock	NUMBER(10)	NOT NULL

4.3 Normalisation

1. **USER (user_id, username, password, user_email, user_address, user_phone_number, registration_date)**

fd1: user_id → username, password, user_email, user_address, user_phone_number, registration_date

1NF&2NF&3NF&BCNF:

USER (user_id, username, password, user_email, user_address, user_phone_number, registration_date)

2. **CART (cart_id, user_id, shipping_id, shipping_method, shipping_address, tracking_no)**

fd1: cart_id → user_id, shipping_id, shipping_method, shipping_address, tracking_no

1NF&2NF&3NF&BCNF:

CART (cart_id, user_id, shipping_id, shipping_method, shipping_address, tracking_no)

3. **SHIPPING (shipping_id, shipping_method, shipping_address, tracking_no)**

fd1: shipping_id → shipping_method, shipping_address, tracking_no

1NF&2NF&3NF&BCNF:

SHIPPING (shipping_id, shipping_method, shipping_address, tracking_no)

4. **ORDER (order_id, product_id, product_quality, unit_price, order_date, delivery_date, total_price, order_status, payment_method_id, shipping_id, customer_id)**

fd1: order_id → product_id, product_quality, unit_price, order_date, delivery_date, total_price, order_status, payment_method_id, shipping_id, customer_id

1NF&2NF&3NF&BCNF:

ORDER (order_id, product_id, product_quality, unit_price, order_date, delivery_date, total_price, order_status, payment_method_id, shipping_id, customer_id)

5. **SELLER (seller_id, shop_name, seller_description, seller_email, seller_rating)**

fd1: seller_id → shop_name, seller_description, seller_email, seller_rating

1NF&2NF&3NF&BCNF:

SELLER (seller_id, shop_name, seller_description, seller_email, seller_rating)

6. **BANK (bank_id, name, password)**

fd1: bank_id → name, password

1NF&2NF&3NF&BCNF:

BANK (bank_id, name, password)

7. **PRODUCT (product_id, name, description, price, category_id, images, seller_id, product_rating)**

fd1: product_id → name, description, price, category_id, images, seller_id, product_rating

1NF&2NF&3NF&BCNF:

PRODUCT (product_id, name, description, price, category_id, images, seller_id, product_rating)

8. **ARVR_INTERACTION (interaction_id, user_id, product_id, interaction_type)**

fd1: interaction_id → user_id, product_id, interaction_type

1NF&2NF&3NF&BCNF:

ARVR_INTERACTION (interaction_id, user_id, product_id, interaction_type)

9. PRODUCT_CATEGORY (product_category_id, product_category_name, product_category_description)

fd1: product_category_id \rightarrow product_category_name, product_category_description

1NF&2NF&3NF&BCNF:

PRODUCT_CATEGORY (product_category_id, product_category_name, product_category_description)

10. PRODUCT_WITH_SIZE (product_id, product_size_id)

fd1: product_id \rightarrow product_size_id

1NF&2NF&3NF&BCNF:

PRODUCT_WITH_SIZE (product_id, product_size_id)

11. PRODUCT_SIZE_DESCRIPTION (product_size_id, product_id, size_name, quantity_in_stock)

fd1: product_size_id \rightarrow product_id, size_name, quantity_in_stock

1NF&2NF&3NF&BCNF:

PRODUCT_SIZE_DESCRIPTION (product_size_id, product_id, size_name, quantity_in_stock)

5.0 Relational DB Schemas (after normalisation)

Set of relation schemas in relational database schema in Vrfify database:

1. USER (user_id, username, password, user_email, user_address, user_phone_number, registration_date)
2. CART (cart_id, user_id, shipping_id, shipping_method, shipping_address, tracking_no)
3. SHIPPING (shipping_id, shipping_method, shipping_address, tracking_no)
4. ORDER (order_id, product_id, product_quality, unit_price, order_date, delivery_date, total_price, order_status, payment_method_id, shipping_id, customer_id)
5. SELLER (seller_id, shop_name, seller_description, seller_email, seller_rating)
6. BANK (bank_id, name, password)
7. PRODUCT (product_id, name, description, price, category_id, images, seller_id, product_rating)
8. ARVR_INTERACTION (interaction_id, user_id, product_id, interaction_type)
9. PRODUCT_CATEGORY (product_category_id, product_category_name, product_category_description)
10. PRODUCT_WITH_SIZE (product_id, product_size_id)
11. PRODUCT_SIZE_DESCRIPTION (product_size_id, product_id, size_name, quantity_in_stock)

USER

user_id	username	password	user_email	user_address	user_phone_number	registration_date
---------	----------	----------	------------	--------------	-------------------	-------------------

CART

cart_id	user_id	shipping_id	shipping_method	shipping_address	tracking_no
---------	---------	-------------	-----------------	------------------	-------------

SHIPPING

shipping_ id	shipping_ method	shipping_ address	tracking_ no
-----------------	---------------------	----------------------	-----------------

ORDER

order_id	product_ id	product_ quality	unit_price	order_date	delivery_ date	total_price
----------	----------------	---------------------	------------	------------	-------------------	-------------

order_ status	payment_method_ id	shipping_ id	customer_id
------------------	-----------------------	-----------------	-------------

SELLER

seller_id	shop_ name	seller_ description	seller_ email	seller_ rating
-----------	---------------	------------------------	------------------	-------------------

BANK

bank_id	name	password
---------	------	----------

PRODUCT

product_id	name	description	price	category_ id	images	seller_id
------------	------	-------------	-------	-----------------	--------	-----------

product_ rating

ARVR INTERACTION

interaction_ id	user_id	product_id	interaction_ _type
--------------------	---------	------------	-----------------------

PRODUCT CATEGORY

product_ category_	product_ category_	product_ category_
-----------------------	-----------------------	-----------------------

id	name	description
----	------	-------------

PRODUCT WITH SIZE

product_id	product_size_id
------------	-----------------

PRODUCT SIZE DESCRIPTION

product_size_id	product_id	size_name	quantity_in_stock
-----------------	------------	-----------	-------------------

6.0 SQL Statements (DDL & DML)

-- DDL

-- Customer Table

```
CREATE TABLE Customer (
    user_id VARCHAR2(15) PRIMARY KEY,
    username VARCHAR2(25) NOT NULL,
    password VARCHAR2(25) NOT NULL,
    user_email VARCHAR2(50) NOT NULL,
    user_address VARCHAR2(100) NOT NULL,
    user_phone_number VARCHAR2(15) NOT NULL,
    registration_date DATE NOT NULL
);
```

-- Seller Table

```
CREATE TABLE Seller(
    seller_id VARCHAR2(25) PRIMARY KEY,
    shop_name VARCHAR2(50) NOT NULL,
    seller_description VARCHAR2(500) NOT NULL,
    seller_phone_number VARCHAR2(15),
    seller_email VARCHAR2(50) NOT NULL,
    seller_rating NUMBER(3,2) NOT NULL
);
```

-- PaymentGateway Table

```
CREATE TABLE PaymentGateway(
    payment_method_id VARCHAR2(25) PRIMARY KEY,
    payment_gateway_name VARCHAR2(25) NOT NULL,
    payment_gateway_description VARCHAR2(150) NOT NULL,
    supported_currencies VARCHAR2(100) NOT NULL
);
```


);

-- Shipping Table

```
CREATE TABLE Shipping(  
    shipping_id VARCHAR2(15) PRIMARY KEY,  
    shipping_method VARCHAR2(25) NOT NULL,  
    tracking_number VARCHAR2(15) NOT NULL,  
    shipping_address VARCHAR2(50) NOT NULL,  
    estimated_delivery_time NUMBER NOT NULL,  
    shipping_cost NUMBER NOT NULL,  
    shipped_from VARCHAR2(50) NOT NULL  
);
```

-- Order Table

```
CREATE TABLE Orders(  
    orders_id VARCHAR2(15) PRIMARY KEY,  
    product_id VARCHAR2(15) NOT NULL,  
    product_quantity NUMBER NOT NULL,  
    unit_price DECIMAL(10, 2) NOT NULL,  
    orders_date DATE NOT NULL,  
    delivery_date DATE,  
    total_price DECIMAL(10, 2) NOT NULL,  
    order_status VARCHAR2(25) NOT NULL,  
    payment_method VARCHAR2(25) NOT NULL,  
    shipping_id VARCHAR2(15),  
    customer_id VARCHAR2(15) NOT NULL,  
    FOREIGN KEY (shipping_id) REFERENCES Shipping (shipping_id),  
    FOREIGN KEY (customer_id) REFERENCES Customer (user_id)  
);
```

-- Product Category Table

```
CREATE TABLE Product_category(  
    product_category_id VARCHAR2(15) PRIMARY KEY,  
    product_category_name VARCHAR2(25) NOT NULL,  
    product_category_description VARCHAR2(500) NOT NULL  
);
```

-- Product Size Table

```
CREATE TABLE Product_with_size(  
    product_size_id VARCHAR2(15) PRIMARY KEY,  
    product_size_name VARCHAR(25) NOT NULL  
);
```

-- Product Table

```
CREATE TABLE Product (  
    product_id VARCHAR2(15) PRIMARY KEY,  
    product_name VARCHAR(50) NOT NULL,  
    product_description VARCHAR(500) NOT NULL,  
    product_price DECIMAL(10, 2) NOT NULL,  
    product_category_id VARCHAR2(15) NOT NULL,  
    product_rating NUMBER NOT NULL,  
    product_images VARCHAR2(300),  
    seller_id VARCHAR2(15) NOT NULL,  
    FOREIGN KEY (seller_id) REFERENCES Seller (seller_id),  
    FOREIGN KEY (product_category_id) REFERENCES Product_category  
(product_category_id)  
);
```

-- Product Size Declaration Table

```
CREATE TABLE Product_size_declaration (  
    size_declaration_id VARCHAR2(15) PRIMARY KEY,  
    product_id VARCHAR2(15) NOT NULL,  
    product_size_id VARCHAR2(15) NOT NULL,  
    quantity_in_stock INTEGER NOT NULL,  
    FOREIGN KEY (product_id) REFERENCES Product (product_id)  
);
```

-- Bank Table

```
CREATE TABLE Bank(  
    bank_id VARCHAR2(15) PRIMARY KEY,  
    bank_name VARCHAR2(50) NOT NULL -- Increased size for bank name  
);
```

-- AR VR Interaction Table

```
CREATE TABLE ARVRInteraction(  
    interaction_id VARCHAR2(15) PRIMARY KEY,  
    user_id VARCHAR2(15) NOT NULL,  
    product_id VARCHAR2(15) NOT NULL,  
    interaction_type VARCHAR2(10) NOT NULL,  
    interaction_timestamp TIMESTAMP,  
    FOREIGN KEY (user_id) REFERENCES Customer (user_id),  
    FOREIGN KEY (product_id) REFERENCES Product (product_id)  
);
```

-- Cart Table

```
CREATE TABLE Cart(  
    cart_id VARCHAR2(15) PRIMARY KEY,
```

```
    user_id VARCHAR2(15) NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES Customer (user_id)  
);
```

-- Cart Product Table

```
CREATE TABLE Cart_products(  
    shopping_cart_id VARCHAR2(15) PRIMARY KEY,  
    product_id VARCHAR2(15) NOT NULL,  
    cart_id VARCHAR2(15) NOT NULL,  
    quantity NUMBER NOT NULL,  
    FOREIGN KEY (cart_id) REFERENCES Cart (cart_id),  
    FOREIGN KEY (product_id) REFERENCES Product (product_id)  
);
```

-- Review Table

```
CREATE TABLE Review(  
    Review_id VARCHAR2(15) PRIMARY KEY,  
    Review VARCHAR2(500) NOT NULL,  
    Review_status VARCHAR2(25) NOT NULL,  
    user_id VARCHAR2(15) NOT NULL,  
    orders_id VARCHAR2(15),  
    FOREIGN KEY (user_id) REFERENCES Customer (user_id),  
    FOREIGN KEY (orders_id) REFERENCES Orders (orders_id)  
);
```

-- Altering Table

```
ALTER TABLE Bank  
ADD bank_balance NUMBER NOT NULL  
ADD user_id VARCHAR2(15) NOT NULL;
```

```
ALTER TABLE Product  
ADD ar_model VARCHAR2(255)  
ADD vr_model VARCHAR2(225)  
ADD product_size_id VARCHAR2(10)  
ADD CONSTRAINT fk_product_size_id_product_with_size FOREIGN KEY  
(product_size_id)  
REFERENCES Product_with_size (product_size_id);
```

```
ALTER TABLE Orders  
ADD OrderNote VARCHAR2(50)  
ADD CourierType VARCHAR2(25) NOT NULL  
ADD arvr_interaction_id VARCHAR2(15)  
ADD order_items VARCHAR2(15) NOT NULL
```

```
ADD order_comment VARCHAR2(100)
ADD payment_method_id VARCHAR2(15)
ADD Review_id VARCHAR2(15)
ADD order_arrived_date DATE
ADD delivery_to VARCHAR2(50)
ADD CONSTRAINT fk_review FOREIGN KEY (Review_id) REFERENCES Review
(Review_id)
ADD CONSTRAINT fk_orders_paymentgateway FOREIGN KEY (payment_method_id)
REFERENCES PaymentGateway (payment_method_id)
ADD CONSTRAINT fk_order_arvr FOREIGN KEY (arvr_interaction_id) REFERENCES
ARVRInteraction (interaction_id)
ADD CONSTRAINT chk_order_status CHECK (order_status IN ('Processing', 'Shipped',
'Completed', 'Cancelled'));
```

```
ALTER TABLE Review
ADD product_quality NUMBER
ADD seller_service NUMBER
ADD delivery_service NUMBER
ADD CONSTRAINT chk_review_status CHECK (Review_status IN ('Submitted',
'Rejected'));
```

```
-----
-----
-- DML
-- Customer
INSERT INTO Customer (user_id, username, password, user_email, user_address,
user_phone_number, registration_date)
VALUES ('C1001', 'Jenny', 'JeNny_24', 'jenny@gmail.com', '21, Jalan Cengal', '0125824252',
TO_DATE('2024-01-16', 'YYYY-MM-DD'));
```

```
INSERT INTO Customer (user_id, username, password, user_email, user_address,
user_phone_number, registration_date)
VALUES('C1002', 'Michael', 'Mikey123', 'michael@gmail.com', '45, Maple Street',
'0157896321', TO_DATE('2024-01-17', 'YYYY-MM-DD'));
```

```
INSERT INTO Customer (user_id, username, password, user_email, user_address,
user_phone_number, registration_date)
VALUES('C1003', 'Emily', 'Emi_567', 'emily@gmail.com', '7, Oak Avenue', '0187456321',
TO_DATE('2024-01-18', 'YYYY-MM-DD'));
```

```
INSERT INTO Customer (user_id, username, password, user_email, user_address,
user_phone_number, registration_date)
VALUES('C1004', 'David', 'DavyD_89', 'david@gmail.com', '12, Birch Lane', '0147852369',
TO_DATE('2024-01-19', 'YYYY-MM-DD'));
```

```
INSERT INTO Customer (user_id, username, password, user_email, user_address,  
user_phone_number, registration_date)  
VALUES('C1005', 'Sophia', 'Sofi123', 'sophia@gmail.com', '33, Cedar Road', '0198745632',  
TO_DATE('2024-01-20', 'YYYY-MM-DD'));
```

-- Seller

```
INSERT INTO Seller (seller_id, shop_name, seller_description, seller_phone_number,  
seller_email, seller_rating)  
VALUES ('S1001', 'Shop Chu', 'Walk-in Shop Chu, where shoes meet enchantment',  
'0158224940', 'shoChuu@gmail.com', 4.5);
```

```
INSERT INTO Seller (seller_id, shop_name, seller_description, seller_phone_number,  
seller_email, seller_rating)  
VALUES('S1002', 'Gadget Galaxy', 'Your one-stop shop for tech wonders', '0123456789',  
'gadgetgalaxy@gmail.com', 4.2);
```

```
INSERT INTO Seller (seller_id, shop_name, seller_description, seller_phone_number,  
seller_email, seller_rating)  
VALUES('S1003', 'Tech Haven', 'Your destination for cutting-edge technology',  
'9876543210', 'techhaven@gmail.com', 4.6);
```

```
INSERT INTO Seller (seller_id, shop_name, seller_description, seller_phone_number,  
seller_email, seller_rating)  
VALUES('S1004', 'Eleele', 'Your go-to destination for cutting-edge gadgets', '0123456789',  
'audioelegance@gmail.com', 4.9);
```

```
INSERT INTO Seller (seller_id, shop_name, seller_description, seller_phone_number,  
seller_email, seller_rating)  
VALUES('S1005', 'Joee', 'Unleash your culinary creativity with our curated collection of  
kitchen cutlery', '018388637', 'Joeee@gmail.com', 5);
```

-- Product Category

```
INSERT INTO Product_category (product_category_id, product_category_name,  
product_category_description)  
VALUES ('CAT001', 'Footwear', 'Shoes and related products');
```

```
INSERT INTO Product_category (product_category_id, product_category_name,  
product_category_description)  
VALUES ('CAT002', 'Electronics', 'Electronic gadgets and devices');
```

```
INSERT INTO Product_category (product_category_id, product_category_name,  
product_category_description)  
VALUES ('CAT003', 'Gadgets and Accessories', 'Electronic gadgets and accessories');
```

```
INSERT INTO Product_category (product_category_id, product_category_name,  
product_category_description)  
VALUES ('CAT004', 'Machines', 'Machines product');
```

```
INSERT INTO Product_category (product_category_id, product_category_name,  
product_category_description)  
VALUES ('CAT005', 'Others', 'Other miscellaneous products');
```


-- Product

```
INSERT INTO Product (product_id, product_name, product_description, product_price,  
product_rating, seller_id, product_category_id, product_images, ar_model, vr_model)  
VALUES ('P1001', 'Charlie & Kathie Woman's High Heels Pumps Shoes', 'High heels from  
Charlie & Kathie', 599.00, 5, 'S1001', 'CAT001', 'image1.jpg', 'ar_model1.obj',  
'vr_model1.obj');
```

```
INSERT INTO Product (product_id, product_name, product_description, product_price,  
product_rating, seller_id, product_category_id, product_images, ar_model, vr_model)  
VALUES ('P1002', 'Samsung Galaxy S21', 'Latest smartphone from Samsung', 799.99, 4.5,  
'S1002', 'CAT002', 'image1.jpg', 'ar_model1.obj', 'vr_model1.obj');
```

```
INSERT INTO Product (product_id, product_name, product_description, product_price,  
product_rating, seller_id, product_category_id, product_images, ar_model, vr_model)  
VALUES ('P1003', 'Canon EOS Rebel T7 DSLR', 'Capture stunning photos with Canon  
DSLR', 599.99, 4.6, 'S1003', 'CAT003', 'image1.jpg', 'ar_model1.obj', 'vr_model1.obj');
```

```
INSERT INTO Product (product_id, product_name, product_description, product_price,  
product_rating, seller_id, product_category_id, product_images, ar_model, vr_model)  
VALUES ('P1004', 'Apple AirPods Pro', 'Immersive audio with Apple AirPods Pro', 249.99,  
4, 'S1004', 'CAT004', 'image1.jpg', 'ar_model1.obj', 'vr_model1.obj');
```

```
INSERT INTO Product (product_id, product_name, product_description, product_price,  
product_rating, seller_id, product_category_id, product_images, ar_model, vr_model)
```

```
VALUES ('P1005', 'Glass Pot', 'Offering a transparent view of your culinary creations as they  
simmer to perfection', 40.00, 4.2, 'S1005', 'CAT005', 'image1.jpg', 'ar_model1.obj',  
'vr_model1.obj');
```

```
-- Declare Product Sizes
```

```
-- Insert into Product_with_size
```

```
INSERT INTO Product_with_size (product_size_id, product_size_name)  
VALUES ('PS1001', '34');
```

```
INSERT INTO Product_with_size (product_size_id, product_size_name)  
VALUES ('PS1002', '35');
```

```
INSERT INTO Product_with_size (product_size_id, product_size_name)  
VALUES ('PS1003', '36');
```

```
INSERT INTO Product_with_size (product_size_id, product_size_name)  
VALUES ('PS1004', '128GB');
```

```
INSERT INTO Product_with_size (product_size_id, product_size_name)  
VALUES ('PS1005', '216GB');
```

```
INSERT INTO Product_with_size (product_size_id, product_size_name)  
VALUES ('PS1006', 'One size');
```

```
-- Match the product with sizes
```

```
INSERT INTO Product_size_declaration (size_declaration_id, product_id, product_size_id,  
quantity_in_stock)  
VALUES ('PSD1001', 'P1001', 'PS1001', 5);
```

```
INSERT INTO Product_size_declaration (size_declaration_id, product_id, product_size_id,  
quantity_in_stock)  
VALUES ('PSD1002', 'P1001', 'PS1002', 5);
```

```
INSERT INTO Product_size_declaration (size_declaration_id, product_id, product_size_id,  
quantity_in_stock)  
VALUES ('PSD1003', 'P1001', 'PS1003', 5);
```

```
INSERT INTO Product_size_declaration (size_declaration_id, product_id, product_size_id,  
quantity_in_stock)  
VALUES ('PSD1004', 'P1002', 'PS1004', 10);
```

```
INSERT INTO Product_size_declaration (size_declaration_id, product_id, product_size_id,
quantity_in_stock)
VALUES ('PSD1005', 'P1002', 'PS1005', 1);
```

```
INSERT INTO Product_size_declaration (size_declaration_id, product_id, product_size_id,
quantity_in_stock)
VALUES ('PSD1006', 'P1003', 'PS1006', 6);
```

```
INSERT INTO Product_size_declaration (size_declaration_id, product_id, product_size_id,
quantity_in_stock)
VALUES ('PSD1007', 'P1004', 'PS1006', 4);
```

```
INSERT INTO Product_size_declaration (size_declaration_id, product_id, product_size_id,
quantity_in_stock)
VALUES ('PSD1008', 'P1005', 'PS1006', 3);
```

```
-- Cart
```

```
INSERT INTO Cart (cart_id, user_id)
VALUES ('CART0001', 'C1001');
```

```
INSERT INTO Cart (cart_id, user_id)
VALUES ('CART0002', 'C1002');
```

```
INSERT INTO Cart (cart_id, user_id)
VALUES ('CART0003', 'C1003');
```

```
INSERT INTO Cart (cart_id, user_id)
VALUES ('CART0004', 'C1004');
```

```
INSERT INTO Cart (cart_id, user_id)
VALUES ('CART0005', 'C1005');
```

```
SELECT * FROM Cart;
```

```
-- Shopping Cart
```

```
INSERT INTO Cart_products (shopping_cart_id, product_id, cart_id, quantity)
VALUES ('SC1001', 'P1001', 'CART0001', 2);
```



```
INSERT INTO Cart_products (shopping_cart_id, product_id, cart_id, quantity)
VALUES ('SC1002', 'P1005', 'CART0001', 2);
```

```
INSERT INTO Cart_products (shopping_cart_id, product_id, cart_id, quantity)
VALUES ('SC1003', 'P1003', 'CART0002', 1);
```

```
INSERT INTO Cart_products (shopping_cart_id, product_id, cart_id, quantity)
VALUES ('SC1004', 'P1004', 'CART0003', 1);
```

```
INSERT INTO Cart_products (shopping_cart_id, product_id, cart_id, quantity)
VALUES ('SC1005', 'P1002', 'CART0004', 3);
```

```
INSERT INTO Cart_products (shopping_cart_id, product_id, cart_id, quantity)
VALUES ('SC1006', 'P1004', 'CART0005', 1);
```

-- Bank for customer

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1001', 'Maybank', 'C1001', 8438.00);
```

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1002', 'Maybank', 'C1002', 2683.43);
```

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1003', 'Bank Islam', 'C1003', 6434.23);
```

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1004', 'Bank Islam', 'C1004', 1239.01);
```

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1005', 'Maybank', 'C1005', 3243.00);
```

-- Bank for seller

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1006', 'Maybank', 'S1001', 10383.00);
```

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1007', 'Maybank', 'S1002', 104834.43);
```

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1008', 'Maybank', 'S1003', 31243.23);
```

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1009', 'CIMB Click', 'S1004', 433224.21);
```

```
INSERT INTO Bank (bank_id, bank_name, user_id, bank_balance)
VALUES ('B1010', 'CIMB Click', 'S1005', 32345.23);
```

```
-- Payment Gateway
```

```
INSERT INTO PaymentGateway (payment_method_id, payment_gateway_name,
payment_gateway_description, supported_currencies)
VALUES ('PG1001', 'FPX', 'Secure online payments', 'USD, EUR, GBP');
```

```
INSERT INTO PaymentGateway (payment_method_id, payment_gateway_name,
payment_gateway_description, supported_currencies)
VALUES ('PG1002', 'Touch n Go', 'Contactless payments and more', 'MYR');
```

```
INSERT INTO PaymentGateway (payment_method_id, payment_gateway_name,
payment_gateway_description, supported_currencies)
VALUES ('PG1003', 'Boost', 'Mobile wallet and payment app', 'MYR');
```

```
INSERT INTO PaymentGateway (payment_method_id, payment_gateway_name,
payment_gateway_description, supported_currencies)
VALUES ('PG1004', 'GrabPay', 'Mobile payments with Grab app', 'MYR, SGD, IDR, THB');
```

```
-- Shipping
```

```
INSERT INTO Shipping (shipping_id, shipping_method, tracking_number,
shipping_address, estimated_delivery_time, shipping_cost, shipped_from)
SELECT 'SHIP1001', 'J&T Express', 'TRACK1001', c.user_address, 3, 4.90, 'J&T Ulu Tiram'
FROM Customer c
WHERE c.user_id = 'C1001';
```

```
SELECT * FROM Shipping;
```

```
INSERT INTO Orders (orders_id, product_id, product_quantity, unit_price, orders_date,
total_price, order_status, payment_method, customer_id, CourierType, order_items,
OrderNote, payment_method_id, order_comment, order_arrived_date, delivery_date,
delivery_to)
SELECT 'ORD1001', cp.product_id, cp.quantity, p.product_price, TO_DATE('2024-01-01',
'YYYY-MM-DD'), cp.quantity * p.product_price + s.shipping_cost, 'Processing', 'FPX',
'C1001', s.shipping_method, 'High heel', 'Plz be careful of my shoes', 'PG1001', 'Order
placed', TO_DATE('2024-01-01', 'YYYY-MM-DD') + s.estimated_delivery_time,
TO_DATE('2024-01-02', 'YYYY-MM-DD'), s.shipping_address
```

```

FROM Cart_products cp
JOIN Product p ON cp.product_id = p.product_id
JOIN Shipping s ON s.shipping_id = 'SHIP1001'
WHERE cp.cart_id = 'CART0001' AND cp.product_id = 'P1001' AND EXISTS (
    SELECT 1 FROM Product_size_declaration psd
    WHERE psd.product_id = cp.product_id AND psd.product_size_id = 'PS1001' AND
    psd.quantity_in_stock >= cp.quantity
);

```

```

SELECT * FROM Orders;

```

```

UPDATE Product_size_declaration
SET quantity_in_stock = quantity_in_stock - (
    SELECT cp.quantity
    FROM Cart_products cp
    WHERE cp.cart_id = 'CART0001' AND cp.product_id = 'P1001' AND product_size_id =
    'PS1001'
)
WHERE product_id = 'P1001' AND product_size_id = 'PS1001';

```

```

SELECT * FROM Product_size_declaration;

```

```

SELECT * FROM Bank;

```

```

-- Update the customer's bank balance

```

```

UPDATE Bank
SET bank_balance = bank_balance - (
    SELECT total_price
    FROM Orders
    WHERE orders_id = 'ORD1001'
)
WHERE user_id = 'C1001';

```

```

-- Update the seller's bank balance

```

```

UPDATE Bank
SET bank_balance = bank_balance + (
    SELECT total_price
    FROM Orders
    WHERE orders_id = 'ORD1001'
)
WHERE user_id = 'S1001';

```

```

SELECT * FROM Bank;

```

```

-- Customer 1 cancel the order
UPDATE Orders
SET order_status = 'Cancelled'
WHERE orders_id = 'ORD1001' AND customer_id = 'C1001';

-- Set order arrival date to NULL
UPDATE Orders
SET order_arrived_date = NULL
WHERE orders_id = 'ORD1001' AND customer_id = 'C1001';

-- Update the customer's bank balance
UPDATE Bank
SET bank_balance = bank_balance + (
    SELECT total_price
    FROM Orders
    WHERE orders_id = 'ORD1001' AND order_status = 'Cancelled'
)
WHERE user_id = 'C1001';

-- Update the seller's bank balance
UPDATE Bank
SET bank_balance = bank_balance - (
    SELECT total_price
    FROM Orders
    WHERE orders_id = 'ORD1001' AND order_status = 'Cancelled'
)
WHERE user_id = 'S1001';

UPDATE Orders
SET order_comment = 'Order has been cancelled'
WHERE orders_id = 'ORD1001' AND order_status = 'Cancelled';

UPDATE Orders
SET delivery_to = NULL
WHERE orders_id = 'ORD1001' AND order_status = 'Cancelled';

SELECT * FROM Bank;

SELECT * FROM Orders;

-----
-- -- Customer 2 view AR and VR model

```

```
SELECT product_id, product_name, product_description, product_price, ar_model,  
vr_model  
FROM Product  
WHERE product_id = 'P1001';
```

```
INSERT INTO Shipping (shipping_id, shipping_method, tracking_number,  
shipping_address, estimated_delivery_time, shipping_cost, shipped_from)  
SELECT 'SHIP1002', 'Ninja', 'TRACK1002', c.user_address, 5, 4.90, 'Ninja Bandar Pulai'  
FROM Customer c  
WHERE c.user_id = 'C1002';
```

```
SELECT * FROM Shipping;
```

-- Order for customer 2 is placed

```
INSERT INTO Orders (orders_id, product_id, product_quantity, unit_price, orders_date,  
total_price, order_status, payment_method, customer_id, CourierType, order_items,  
OrderNote, payment_method_id, order_comment, order_arrived_date, delivery_date,  
delivery_to)  
SELECT 'ORD1002', cp.product_id, cp.quantity, p.product_price, TO_DATE('2024-01-01',  
'YYYY-MM-DD'), cp.quantity * p.product_price + s.shipping_cost, 'Processing', 'FPX',  
'C1002', 'Ninja', 'Camera', 'Plz be more careful, inside is camera.', 'PG1001', 'Order placed',  
TO_DATE('2024-01-01', 'YYYY-MM-DD') + s.estimated_delivery_time,  
TO_DATE('2024-01-02', 'YYYY-MM-DD'), s.shipping_address  
FROM Cart_products cp  
JOIN Product p ON cp.product_id = p.product_id  
JOIN Shipping s ON s.shipping_id = 'SHIP1002'  
WHERE cp.cart_id = 'CART0002' AND cp.product_id = 'P1003' AND EXISTS (  
    SELECT 1 FROM Product_size_declaration psd  
    WHERE psd.product_id = cp.product_id AND psd.quantity_in_stock >= cp.quantity  
);
```

```
UPDATE Product_size_declaration  
SET quantity_in_stock = quantity_in_stock - (  
    SELECT cp.quantity  
    FROM Cart_products cp  
    WHERE cp.cart_id = 'CART0002' AND cp.product_id = 'P1003'  
)  
WHERE product_id = 'P1003';
```

```
SELECT * FROM Product_size_declaration;
```

-- Update the customer's bank balance

```
UPDATE Bank  
SET bank_balance = bank_balance - (
```

```

SELECT total_price
FROM Orders
WHERE orders_id = 'ORD1002'
)
WHERE user_id = 'C1002';

-- Update the seller's bank balance
UPDATE Bank
SET bank_balance = bank_balance + (
    SELECT total_price
    FROM Orders
    WHERE orders_id = 'ORD1002'
)
WHERE user_id = 'S1002';

-- Update the order's has been shipped
UPDATE Orders
SET order_status = 'Shipped'
WHERE orders_id = 'ORD1002' AND customer_id = 'C1002';

-- Update the order's has been completed
UPDATE Orders
SET order_status = 'Completed'
WHERE orders_id = 'ORD1002' AND customer_id = 'C1002';

UPDATE Orders
SET order_comment = 'Order has been out of delivery'
WHERE orders_id = 'ORD1002' AND order_status = 'Completed';

-- Allow customer to add a review
-- Fake review
INSERT INTO Review (Review_id, Review, user_id, Review_status, orders_id,
product_quality, seller_service, delivery_service)
VALUES ('R1001', 'My meow so kiut', 'C1002', 'Rejected', 'ORD1002', 5, 5, 5);

SELECT * FROM Review;

-- Update the Orders table with the new Review_id
UPDATE Orders
SET Review_id = 'R1001'
WHERE Review_id = 'R1001';

-- Real review

```

```
UPDATE Review
SET Review = 'This camera is in good condition!'
WHERE Review_id = 'R1001';
```

```
UPDATE Review
SET Review_status = 'Submitted'
WHERE Review_id = 'R1001';
```

```
-- Retrieve the review for a specific order
SELECT o.orders_id, r.Review
FROM Orders o
JOIN Review r ON o.Review_id = r.Review_id
WHERE o.orders_id = 'ORD1002' AND o.customer_id = 'C1002';
```

```
SELECT * FROM Product_with_size;
```

```
SELECT * FROM Product_size_declaration;
```

```
SELECT * FROM Seller;
```

```
SELECT * FROM Product;
```

```
SELECT * FROM Orders;
```

```
SELECT * FROM Bank;
```

```
SELECT * FROM Review;
```

7.0 Summary

In conclusion, we have updated the Conceptual ERD, business rules, database conceptual design, and implemented the SQL statements for the database throughout this phase. With the addition of cutting-edge augmented reality (AR) and virtual reality (VR) technologies, this enhanced system will bring in a new age of rich and immersive e-commerce experiences. Analysing functional connections between relationships under the guidance of the updated business rules was a crucial step from conceptual Entity-Relationship Diagrams (ERD) to logical ERDs. At the same time, modifications were made to the data dictionary by using normalised relations that evolved from the adjustments made to the logical ERD. The updated data dictionary is important as it provides detailed information about all entities, properties, and relationships. Last but not least, we used Oracle Apex to execute the SQL statement to create the tables. This stage is to make sure that the database is in alignment with our main goal which is to improve the e-commerce experience on the "Vrify" platform. With the integration of AR and VR features, the e-commerce experience is elevated to a new dimension, offering customers immersive product interactions and a visually enriched shopping journey.