Assumptions:

1. 1 customer can have many customer addresses, but at least one address
2. 1 customer can have many orders
3. 1 delivery match with 1 customer address
4. 1 order only matches with 1 transaction, 1 transaction match with 1 delivery
5. 1 order match with 1 order details
6. 1 order can obtain many products, and the sub\_quantity shows the number of each product separately
7. 1 supplier can supply many products
8. 1 product belongs to 1 category, and 1 category can have many different products
9. In category,, we have different layers, for example, the parent category is clothes, category under clothes can be shirts, tops, bottoms etc.
10. 1 product can have many ads or 1 ad or 0 ad.
11. If the delivery status is ‘ not delivered’, the delivery start date and end date are empty. If the delivery status is ‘In\_Delivery’, the delivery start date is not empty but the delivery end date is empty. If the delivery status is ‘Completed’, the delivery start date and end date can’t be empty. If the delivery status is ‘Failed’, the start date and end date are not empty.
12. If the transaction status is ‘Pending’,’Processing’ or ‘Cancelled’, there is no delivery. If the transaction status is ‘Succeed’, it matches with one delivery record.
13. If the customer is membership, then we will show it with discount price
14. The uni\_price in order is match with product price or discount price, depends on the customer is membership or not
15. If the order status is ‘Pending’,’Processing’ or ‘Cancelled’, there is no transaction matched with it. If the order status is ‘Succeed’, it matches with one transaction record.
16. The delivery date is from 2023-09-01 to 2024-03-01, and also if the delivery data is more than 30 days, then the delivery status will showed as ‘Failed’.
17. The order date is from 2023-08-01 to 2024-03-01
18. The transaction time is depend on the order date, for example if the order date is 2024-01-01, the transaction time is same date but with specific time 2024-01-01+ h-m-s

CREATE TABLE customer (

customer\_id INT PRIMARY KEY NOT NULL,

first\_name VARCHAR (50) NOT NULL,

last\_name VARCHAR (50) NOT NULL,

gender ENUM (‘male’, 'female’,’other’) NOT NULL,

membership ENUM (‘yes’, ‘no’) NOT NULL,

date\_of\_birth DATE NOT NULL,

email VARCHAR (255) NOT NULL UNIQUE,

password\_hash VARCHAR (255) NOT NULL,

);

CREATE TABLE customer\_address (

customer\_address\_id INT PRIMARY KEY NOT NULL,

customer\_id INT NOT NULL,

zip\_code CHAR (10) NOT NULL,

country CHAR (20) NOT NULL,

state VARCHAR (50) NOT NULL,

city VARCHAR (50) NOT NULL,

street VARCHAR (255) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id)

);

CREATE TABLE orders (

order\_id INT PRIMARY KEY NOT NULL,

product\_id INT NOT NULL,

customer\_id INT NOT NULL,

order\_time TIMESTAMP NOT NULL,

order\_status ENUM(‘Pending’, ‘Processing’, ‘Succeed’, ‘Cancelled’) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id),

FOREIGN KEY (product\_id) REFERENCES product (product\_id)

);

CREATE TABLE order\_details (

order\_detail\_id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

order\_id INT NOT NULL,

product\_id INT NOT NULL,

sub\_quantity INT NOT NULL,

FOREIGN KEY (order\_id) REFERENCES orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES product(product\_id)

);

CREATE TABLE transaction (

transaction\_id INT PRIMARY KEY NOT NULL,

order\_id INT NOT NULL,

customer\_id INT NOT NULL,

transaction\_time TIMESTAMP,

payment\_method ENUM (‘Credit card’, ‘Debit card’, ‘Paypal’) NOT NULL,

transaction\_status ENUM (‘Pending’, ‘Processing’, ‘Succeed’, ’Cancelled’) NOT NULL,

FOREIGN KEY (order\_id) REFERENCES order (order\_id),

FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id)

);

CREATE TABLE delivery (

delivery\_id INT PRIMARY KEY NOT NULL,

transaction\_id INT NOT NULL,

customer\_address\_id INT NOT NULL,

delivery\_status ENUM (‘Not\_Deliveried’, ‘In\_Delivery’, ‘Completed’, ‘Failed’) NOT NULL,

delivery\_start\_date DATE NOT NULL,

delivery\_end\_date DATE NOT NULL,

FOREIGN KEY (transaction\_id) REFERENCES transaction (transaction\_id),

FOREIGN KEY (customer\_address\_id) REFERENCES customer\_address (customer\_address\_id)

);

CREATE TABLE product (

product\_id INT PRIMARY KEY NOT NULL,

category\_id INT NOT NULL,

product\_name VARCHAR (100) NOT NULL,

product\_price DECIMAL (10,2) NOT NULL,

discount\_percent FLOAT NOT NULL,

rate\_value DECIMAL (2, 1) CHECK (rate\_value >= 0 AND rate\_value <= 5),

FOREIGN KEY (category\_id) REFERENCES category (category \_id)

);

CREATE TABLE category (

category\_id INT PRIMARY KEY NOT NULL,

parent\_category\_id INT,

category\_type VARCHAR (50) NOT NULL,

FOREIGN KEY (parent\_category\_id) REFERENCES category (category\_id)

);

CREATE TABLE supply (

supplier\_id INT NOT NULL,

product\_id INT NOT NULL,

stock INT NOT NULL,

PRIMARY KEY (supplier\_id, product\_id),

FOREIGN KEY (supplier\_id) REFERENCES supplier (supplier\_id),

FOREIGN KEY (product\_id) REFERENCES product (product\_id)

) ;

CREATE TABLE supplier (

supplier\_id INT PRIMARY KEY NOT NULL,

supplier\_phone VARCHAR (20) NOT NULL,

supplier\_name VARCHAR (100) NOT NULL,

supplier\_email VARCHAR (255) NOT NULL UNIQUE,

supplier\_zip\_code CHAR (10) NOT NULL,

supplier\_country CHAR (20) NOT NULL,

supplier\_state VARCHAR (50) NOT NULL,

supplier\_city VARCHAR (50) NOT NULL,

supplier\_street VARCHAR (255) NOT NULL

) ;

CREATE TABLE ads (

ad\_id INT PRIMARY KEY NOT NULL,

product\_id INT NOT NULL,

ad\_status ENUM(‘active’, ‘ended’),

ad\_start\_date DATE NOT NULL,

ad\_end\_date DATE NOT NULL,

FOREIGN KEY (product\_id) REFERENCES product (product\_id)

);