Kevin Ortiz

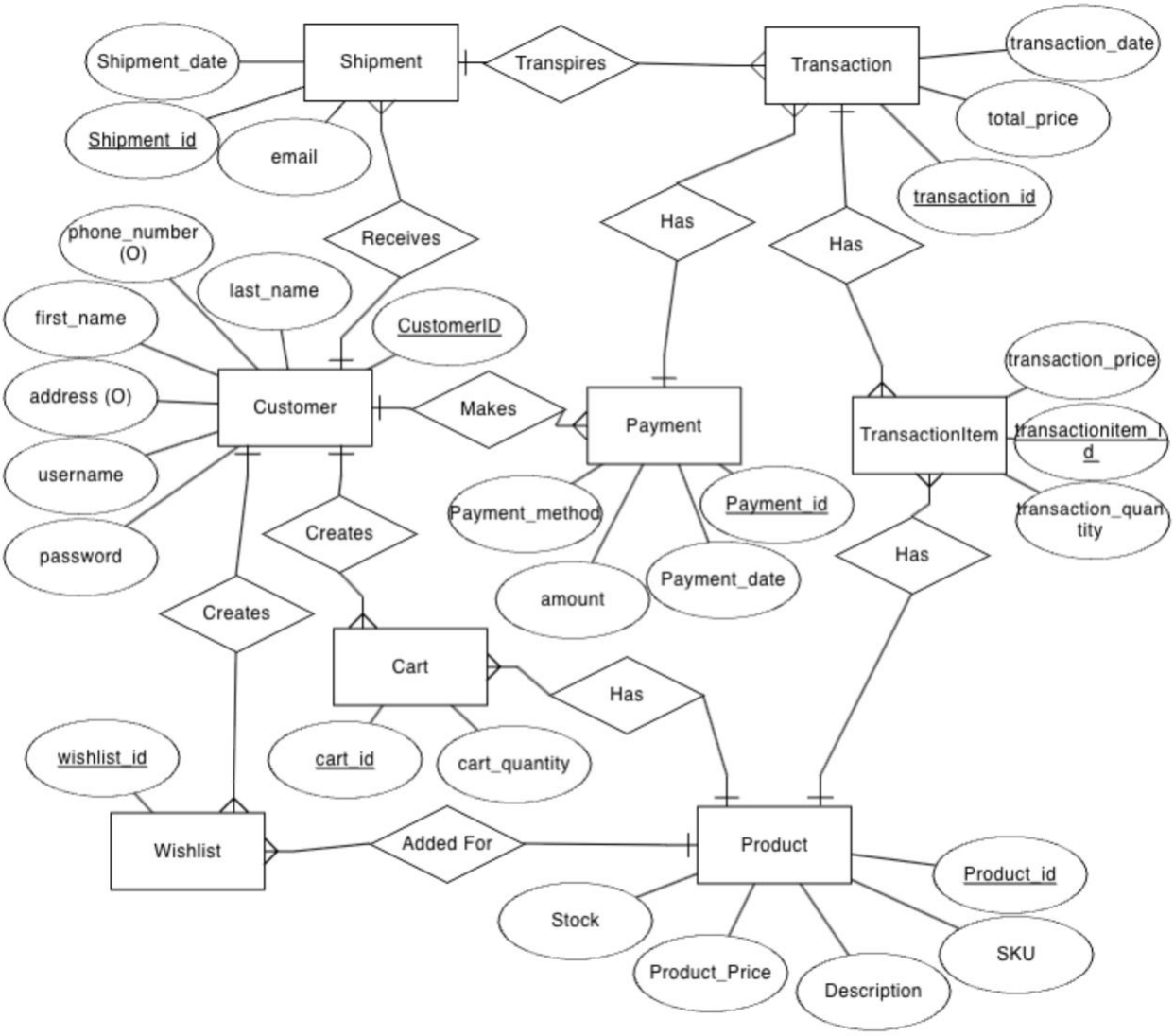
07-02-2023

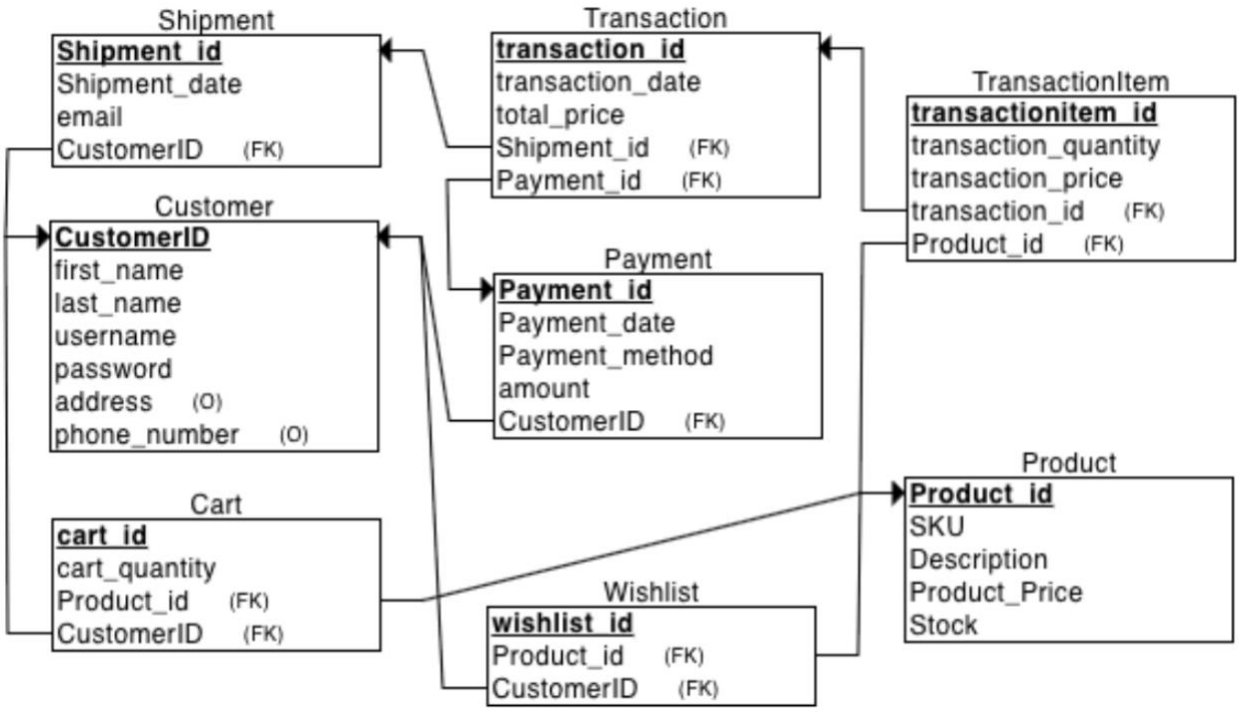
Client Project: Ecommerce Startup for eBooks

Executive Summary

In the future, I hope to make a digital store that provides digital products like eBooks or online courses (in part to simplify logistics as my first startup). Since the industry is so tech-based, the best companies, without a doubt, must leverage data as a competitive strategy against other platforms. As a result, the best data would be anything used for retaining reoccurring customers or measuring the success of the products. The proposed startup here will have many of the same foundations as a standard e-commerce store, but its core objective will be to focus on improving repeat purchases as the core objective (as a vital KPI for product quality). By prioritizing product value and customer loyalty, the startup aims to foster long-term continual super fans; however, it can only be leveraged correctly with a solid database.

The database framework & architecture in the following pages was made as a solid framework that could be easily scaled no matter how much data it could have. The possible data will look like the data of any average e-commerce store but is done to help answer the questions aligned with the core objective, but agile enough to ask new questions if the circumstances call for it. Overall, the database solution outlined in the following pages will serve as a solid foundation for an e-commerce startup focusing on driving repeat purchases (and product quality). With a scalable and agile database framework in place, the startup will be well-positioned to leverage the database's capabilities to its total capacity.


Shipment_date
Shipment
Transpires
Transaction
transaction_date
total_price
Shipment_id
email
transaction_id
Has
Has
Receives
phone_number (O)
last_name
first_name
CustomerID
address (O)
Customer
Makes
Payment
TransactionItem
transaction_price
username
Payment_method
Payment_id
password
Creates
amount
Payment_date
Has
Creates
Cart
Has
wishlist_id
cart_id
cart_quantity
Wishlist
Added For
Product
Product_id
Stock
SKU
Product_Price
Description


Shipment
Shipment_id
Shipment_date
email
CustomerID (FK)
Transaction
transaction_id
transaction_date
total_price
Shipment_id (FK)
Payment_id (FK)
TransactionItem
transactionitem_id
transaction_quantity
transaction_price
transaction_id (FK)
Product_id (FK)
Customer
CustomerID
first_name
last_name
username
password
address (O)
phone_number (O)
Payment
Payment_id
Payment_date
Payment_method
amount
CustomerID (FK)
Product
Product_id
SKU
Description
Product_Price
Stock
Cart
cart_id
cart_quantity
Product_id (FK)
CustomerID (FK)
Wishlist
wishlist_id
Product_id (FK)
CustomerID (FK)

SHIPMENT

Shipment_id	Shipment_date	email	CustomerID
1	6/1/2020	John.Doe@email.com	1
2	6/2/2020	Jane.Smith@email.com	2
3	6/3/2020	John.Doe@email.com	1
4	6/4/2020	Michael Johnson@email.com	3
5	6/5/2020	Jane.Smith@email.com	2

CUSTOMER

CustomerID	first_name	last_name	username	password	address	phone_number
1	John	Doe	johndoe	abc1	111 Main St.	3034567890
2	Jane	Smith	janesmith	pass	222 Ave.	3036543210
3	Michael	Johnson	mjohnson	a1234	333 Blvd.	3031234567
4	Emily	Brown	ebrown	abcd	444 Road	3039990000
5	David	Wilson	dwilson	qw12	555 Park Ave	3035556666

TRANSACTION

transaction_id	transaction_date	total_price	Shipment_id	Payment_id
1	6/1/2020	50	1	1
2	6/2/2020	30	2	2
3	6/3/2020	80	3	3
4	6/4/2020	65	4	4
5	6/5/2020	45	5	5

PAYMENT

Payment_id	Payment_date	Payment_method	Amount	CustomerID
1	6/1/2023	Credit Card	50	1
2	6/2/2023	PayPal	30	2
3	6/3/2023	Debit Card	80	1
4	6/4/2023	Cash	65	4
5	6/5/2023	PayPal	45	3

PRODUCT

Product_id	SKU	Description	Product_Price	Stock
1	SKU01	EBook - How to Fail: 2023 ver.	20	10
2	SKU02	EBook - The Art of Procrastinati	15	5
3	SKU03	EBook - The Secret Life of Sock	10	8
4	SKU04	EBook - Lazy Guide to Producti	25	3
5	SKU05	EBook - Cooking for Dummies	12	12

```
CREATE TABLE Customer
(
  CustomerID INT NOT NULL,
  first_name VARCHAR(255) NOT NULL,
  last_name VARCHAR(255) NOT NULL,
  username VARCHAR(255) NOT NULL,
  password VARCHAR(255) NOT NULL,
  address VARCHAR(255),
  phone_number INT,
  PRIMARY KEY (CustomerID)
);

CREATE TABLE Payment
(
  Payment_id INT NOT NULL,
  Payment_date DATE NOT NULL,
  Payment_method VARCHAR(255) NOT NULL,
  amount INT NOT NULL,
  CustomerID INT NOT NULL,
  PRIMARY KEY (Payment_id),
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

CREATE TABLE Product
(
  Product_id INT NOT NULL,
  SKU VARCHAR(255) NOT NULL,
  Description VARCHAR(255) NOT NULL,
  Product_Price INT NOT NULL,
  Stock INT NOT NULL,
  PRIMARY KEY (Product_id)
);

CREATE TABLE Wishlist
(
  wishlist_id INT NOT NULL,
  Product_id INT NOT NULL,
  CustomerID INT NOT NULL,
  PRIMARY KEY (wishlist_id),
  FOREIGN KEY (Product_id) REFERENCES Product(Product_id),
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

CREATE TABLE Shipment
(
  Shipment_id INT NOT NULL,
  Shipment_date DATE NOT NULL,
  email VARCHAR(255) NOT NULL,
  CustomerID INT NOT NULL,
  PRIMARY KEY (Shipment_id),
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

CREATE TABLE Transaction
(
  transaction_id INT NOT NULL,
  transaction_date DATE NOT NULL,
  total_price INT NOT NULL,
  Shipment_id INT NOT NULL,
  Payment_id INT NOT NULL,
  PRIMARY KEY (transaction_id),
  FOREIGN KEY (Shipment_id) REFERENCES Shipment(Shipment_id),
  FOREIGN KEY (Payment_id) REFERENCES Payment(Payment_id)
);

CREATE TABLE TransactionItem
(
  transactionitem_id INT NOT NULL,
  transaction_quantity INT NOT NULL,
  transaction_price INT NOT NULL,
  transaction_id INT NOT NULL,
  Product_id INT NOT NULL,
  PRIMARY KEY (transactionitem_id),
  FOREIGN KEY (transaction_id) REFERENCES Transaction(transaction_id),
  FOREIGN KEY (Product_id) REFERENCES Product(Product_id)
```

```
);

CREATE TABLE Cart
(
  cart_id INT NOT NULL,
  cart_quantity INT NOT NULL,
  Product_id INT NOT NULL,
  CustomerID INT NOT NULL,
  PRIMARY KEY (cart_id),
  FOREIGN KEY (Product_id) REFERENCES Product(Product_id),
  FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

/* Ecommerce Startup for eBooks – INSERT INTO statements*/

INSERT INTO CUSTOMER VALUES (1, 'John', 'Doe', 'johndoe', 'abc1', '111 Main St.', '3034567890');
INSERT INTO CUSTOMER VALUES (2, 'Jane', 'Smith', 'janesmith', 'pass', '222 Ave.', '3036543210');
INSERT INTO CUSTOMER VALUES (3, 'Michael', 'Johnson', 'mjohnson', 'a1234', '333 Blvd.', '3031234567');
INSERT INTO CUSTOMER VALUES (4, 'Emily', 'Brown', 'ebrown', 'abcd', '444 Road', '3039990000');
INSERT INTO CUSTOMER VALUES (5, 'David', 'Wilson', 'dwilson', 'qw12', '555 Park Ave.', '3035556666');

INSERT INTO SHIPMENT VALUES (1, TO_DATE('6/1/2020', 'MM/DD/YYYY'), 'John.Doe@email.com', 1);
INSERT INTO SHIPMENT VALUES (2, TO_DATE('6/2/2020', 'MM/DD/YYYY'), 'Jane.Smith@email.com', 2);
INSERT INTO SHIPMENT VALUES (3, TO_DATE('6/3/2020', 'MM/DD/YYYY'), 'John.Doe@email.com', 1);
INSERT INTO SHIPMENT VALUES (4, TO_DATE('6/4/2020', 'MM/DD/YYYY'), 'Michael Johnson@email.com', 3);
INSERT INTO SHIPMENT VALUES (5, TO_DATE('6/5/2020', 'MM/DD/YYYY'), 'Jane.Smith@email.com', 2);

INSERT INTO Payment VALUES (1, TO_DATE('6/1/2023', 'MM/DD/YYYY'), 'Credit Card', 50, 1);
INSERT INTO Payment VALUES (2, TO_DATE('6/2/2023', 'MM/DD/YYYY'), 'PayPal', 30, 2);
INSERT INTO Payment VALUES (3, TO_DATE('6/3/2023', 'MM/DD/YYYY'), 'Debit Card', 80, 1);
INSERT INTO Payment VALUES (4, TO_DATE('6/4/2023', 'MM/DD/YYYY'), 'Cash', 65, 4);
INSERT INTO Payment VALUES (5, TO_DATE('6/5/2023', 'MM/DD/YYYY'), 'PayPal', 45, 3);

INSERT INTO Product VALUES (1, 'SKU01', 'EBook - How to Fail: 2023 ver.', 20, 10);
INSERT INTO Product VALUES (2, 'SKU02', 'EBook - The Art of Procrastination', 15, 5);
INSERT INTO Product VALUES (3, 'SKU03', 'EBook - The Secret Life of Socks', 10, 8);
INSERT INTO Product VALUES (4, 'SKU04', 'EBook - Lazy Guide to Productivity', 25, 3);
INSERT INTO Product VALUES (5, 'SKU05', 'EBook - Cooking for Dummies', 12, 12);

INSERT INTO Cart VALUES (1, 2, 1, 1);
INSERT INTO Cart VALUES (2, 1, 2, 2);
INSERT INTO Cart VALUES (3, 3, 4, 1);
INSERT INTO Cart VALUES (4, 2, 3, 3);
INSERT INTO Cart VALUES (5, 1, 5, 2);

INSERT INTO Wishlist VALUES ('1', '2', '1');
INSERT INTO Wishlist VALUES ('2', '4', '2');
INSERT INTO Wishlist VALUES ('3', '1', '3');
INSERT INTO Wishlist VALUES ('4', '3', '1');
INSERT INTO Wishlist VALUES ('5', '5', '4');

INSERT INTO TRANSACTION VALUES (1, TO_DATE('06/01/2020', 'MM/DD/YYYY'), 50, 1, 1);
INSERT INTO TRANSACTION VALUES (2, TO_DATE('6/2/2020', 'MM/DD/YYYY'), 30, 2, 2);
INSERT INTO TRANSACTION VALUES (3, TO_DATE('6/3/2020', 'MM/DD/YYYY'), 80, 3, 3);
INSERT INTO TRANSACTION VALUES (4, TO_DATE('6/4/2020', 'MM/DD/YYYY'), 65, 4, 4);
INSERT INTO TRANSACTION VALUES (5, TO_DATE('6/5/2020', 'MM/DD/YYYY'), 45, 5, 5);

INSERT INTO TransactionItem VALUES (1, 2, 40, 1, 1);
INSERT INTO TransactionItem VALUES (2, 1, 15, 2, 2);
INSERT INTO TransactionItem VALUES (3, 3, 75, 3, 4);
INSERT INTO TransactionItem VALUES (4, 2, 20, 4, 3);
INSERT INTO TransactionItem VALUES (5, 1, 12, 5, 5);
```

Questions to answer:

Who are repeat customers?

```sql
SELECT c.CustomerID, c.last_name, c.first_name
FROM Customer c
INNER JOIN Payment t ON c.CustomerID = t.customerid
GROUP BY c.CustomerID, c.last_name, c.first_name
HAVING COUNT(t.Payment_id) >1;
```

Which customers have not yet made another purchase? (Quantity of orders >2)

```sql
SELECT c.CustomerID, c.last_name, c.first_name
FROM Customer c
WHERE (
    SELECT COUNT (*)
    FROM Payment t
    WHERE t.CustomerID = c.CustomerID
) <=2;
```