Phase 2

CS461-DATABASE SYSTEMS PROJECT

Sara Sutton

Zachary Warunek and Nash Reigle

SQL Statements

**Product**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PID** | **Name** | **Description** | **Seller** | **Price** | **Prime Eligible** |

CREATE TABLE Product (

PID INTEGER IDENTITY(1,1) PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

Description VARCHAR(255) NOT NULL,

Seller VARCHAR(255) NOT NULL,

Price DECIMAL(18, 2) NOT NULL,

Prime\_Eligible BIT NOT NULL

);

The product ID is an incrementing counter. Product ID would be the primary key. Unable to be deleted as the posted product will be referenced in purchase history.

**Account**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Account ID** | **First Name** | **Last Name** | **Prime Member** | **Password** |

CREATE TABLE Account (

AccountID INTEGER IDENTITY(1,1) PRIMARY KEY,

First Name VARCHAR(255) NOT NULL,

Last Name VARCHAR(255) NOT NULL,

Password VARCHAR(255) NOT NULL,

Prime\_Member BIT NOT NULL

);

The customers would need multiple forms of information on file. The ID could either be a string or a number, the first name, last name, address, and password would be strings, and the Prime membership could be stored as a date, seeing as a membership is a subscription and not a one time purchase. The customer ID would be the primary key as it would need to be unique and other tables depend on the ID for products and creating subtables such as the payment methods

The Account IDs will start at 1 and increment upwards to prevent repeats or overlaps. The rest of the data can be identical to another as names and passwords can be identical with no correlation.

**Payment Type**

|  |  |
| --- | --- |
| **PTID** | **Type** |

CREATE TABLE Payment\_Type (

PTID INTEGER IDENTITY(1,1) PRIMARY KEY,

PTType VARCHAR(255) NOT NULL

);

This will be a set list of valid payment types to be used in the payment method table

The Payment Type ID is the primary Key of this table and increments upward.

**Payment Method**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Account ID** | **PMID** | **PTID** | **NameOnCard** | **PMCardNumber** | **PMExp** | **PMCVV** | **PMFavorite** |

CREATE TABLE Payment\_Method (

PMID INTEGER IDENTITY(1,1) PRIMARY KEY,

AccountID INTEGER FOREIGN KEY REFERENCES Account(AccountID),

PTID INTEGER FOREIGN KEY REFERENCES Payment\_Type(PTID),

NameOnCard VARCHAR(255) NULL,

CardNumber VARCHAR(16) NULL,

PMExp VARCHAR(5) NULL,

PMCvv VARCHAR(3) NULL,

PMFavorite BIT NOT NULL

);

The payment method would store information such as the credit card number or account id for digital payments, the card expiration and CVV if applicable, and a binary option if the card is the favorite to default to that option. The information will be stored as an ID for easier referencing and access that would allow for deletion of payment methods.

The Payment Method is the primary key of this table and increments upward. The Payment Method Account ID refers to the Account ID. The Payment Method PTID refers to the Payment Type PTID. The rest of the variables are open due to the variances of payment methods such as credit cards, debit cards, paypal and bitcoin.

**Address**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AccountID** | **AddressID** | **Address** | **Street** | **City** | **State** | **Country** | **Zip** |

CREATE TABLE Address (

AddressID INTEGER IDENTITY(1,1) PRIMARY KEY,

AccountID INTEGER FOREIGN KEY REFERENCES Account(AccountID),

Address VARCHAR(255) NOT NULL,

Street VARCHAR(255) NOT NULL,

City VARCHAR(255) NOT NULL,

State VARCHAR(255) NOT NULL,

Country VARCHAR(255) NOT NULL,

Zip VARCHAR(255) NOT NULL,

);

A unique ID is created for each address and is labeled the primary key. The Account ID in address references the AccountID in Account. Multiple addresses can be stored this way.

**Watchlist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Account ID** | **Name** | **Description** | **Total** | **PLID** |

CREATE TABLE Watchlist (

Name VARCHAR(255) PRIMARY KEY,

AccountID INTEGER FOREIGN KEY REFERENCES Account(AccountID),

Descrition VARCHAR(255) NOT NULL,

Total INTEGER NOT NULL,

PLID INTEGER FOREIGN KEY REFERENCES Product\_List(PLID),

);

AccountID. The watchlist can hold the total of the products and the description of those products given by the users, such as “this would be a good birthday present for \_\_\_\_”. The rest of the information of the can be grabbed as PLID is referenced to the Product List. The watchlist is created and named by the user, and the user is referenced by their account’s

**Recommendation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Account ID** | **PID** | **Relevance** | **Tags** | **NumberOfVeiws** |

CREATE TABLE Recommendation (

PID INTEGER FOREIGN KEY REFERENCES Product(PID),

AccountID INTEGER FOREIGN KEY REFERENCES Account(AccountID),

Relevance INTEGER NOT NULL,

Tags VARCHAR(255) NOT NULL,

Views INTEGER NOT NULL,

);

The recommendations will be a table that is used for generating recommended products. The recommendation type will determine if the recommendation tag is a tag or a seller, and the number of times viewed counter will increase if a person visits or buys a product. In order to keep things relevant to the consumer, the table has a “most recently accessed” date, so a tag with less hits but was viewed recently will rank higher for recommendations. The consumer will be able to say “I am not interested in \_\_\_” to fine tune their recommendations or sellers.

For the Recommendation table, the Recommendation PID refers to the Product PID. The AccountID in recommendation refers to the AccountID in Account

**Shopping Cart**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Account ID** | **SCID** | **Total** | **Prime Eligible** | **PLID** |

CREATE TABLE Shopping\_Cart (

SCID INTEGER IDENTITY(1,1) PRIMARY KEY,

AccountID INTEGER FOREIGN KEY REFERENCES Account(AccountID),

Total INTEGER NOT NULL,

Prime\_Eligible BIT NOT NULL,

PLID INTEGER FOREIGN KEY REFERENCES Product\_List(PLID),

);

The shopping cart will hold information about individual products that a customer has selected, the quanity, the price, any applicable deals, and their eligibility for prime.

A unique shopping cart entry is created for each item added to the shopping cart that is labeled with a foreign key AccountID that refers to the AccountID in Account. The PLID refers to the Product List PLID.

**Product List**

|  |  |  |  |
| --- | --- | --- | --- |
| **Account ID** | **PLID** | **PID** | **Quantity** |

CREATE TABLE Product\_List (

PLID INTEGER IDENTITY(1,1) PRIMARY KEY,

AccountID INTEGER FOREIGN KEY REFERENCES Account(AccountID),

PID INTEGER FOREIGN KEY REFERENCES Product(PID),

Quantity INTEGER NOT NULL

);

The Product List holds the table of items that a user has purchased, added to their watchlist, or put in their shopping cart. The PLID is the primary key and AccountID is referred from the Account table. The PID is referred from the Product table, which holds information such as price, discounts, name, and prime eligibility.

**Product Order**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AccountID** | **Order Number** | **Status** | **Total** | **PLID** | **AddressID** | **PMID** | **Timestamp** |

CREATE TABLE Product\_Order(

OrderNumber INTEGER IDENTITY(1,1) PRIMARY KEY,

AccountID INTEGER FOREIGN KEY REFERENCES Account(AccountID),

Status TINYINT NOT NULL,

Total INTEGER NOT NULL,

Time TIMESTAMP NOT NULL,

PLID INTEGER FOREIGN KEY REFERENCES Product\_List(PLID),

AddressID INTEGER FOREIGN KEY REFERENCES Address(AddressID),

PMID INTEGER FOREIGN KEY REFERENCES Payment\_Method(PMID)

);

Each order number is unique and the primary key of this table. The Account that purchased the order is attached to this table from the Account Table. The PLID is referred from the Product List PLID and the AddressID is referred from the Address table. The PMID tracks the payment method from the Payment Method table.

The status marks the order as pending, received, inprocess, shipped, and delivered.

The total is pulled from the current(at the time of order placement) prices/discounts of the products from the Product table by the PID through the Product List PLID.

**Viewed**

|  |  |  |  |
| --- | --- | --- | --- |
| Account ID | PID | Times Viewed | LastViewed |

CREATE TABLE Viewed (

PID INTEGER FOREIGN KEY REFERENCES Product(PID),

AccountID INTEGER FOREIGN KEY REFERENCES Account(AccountID),

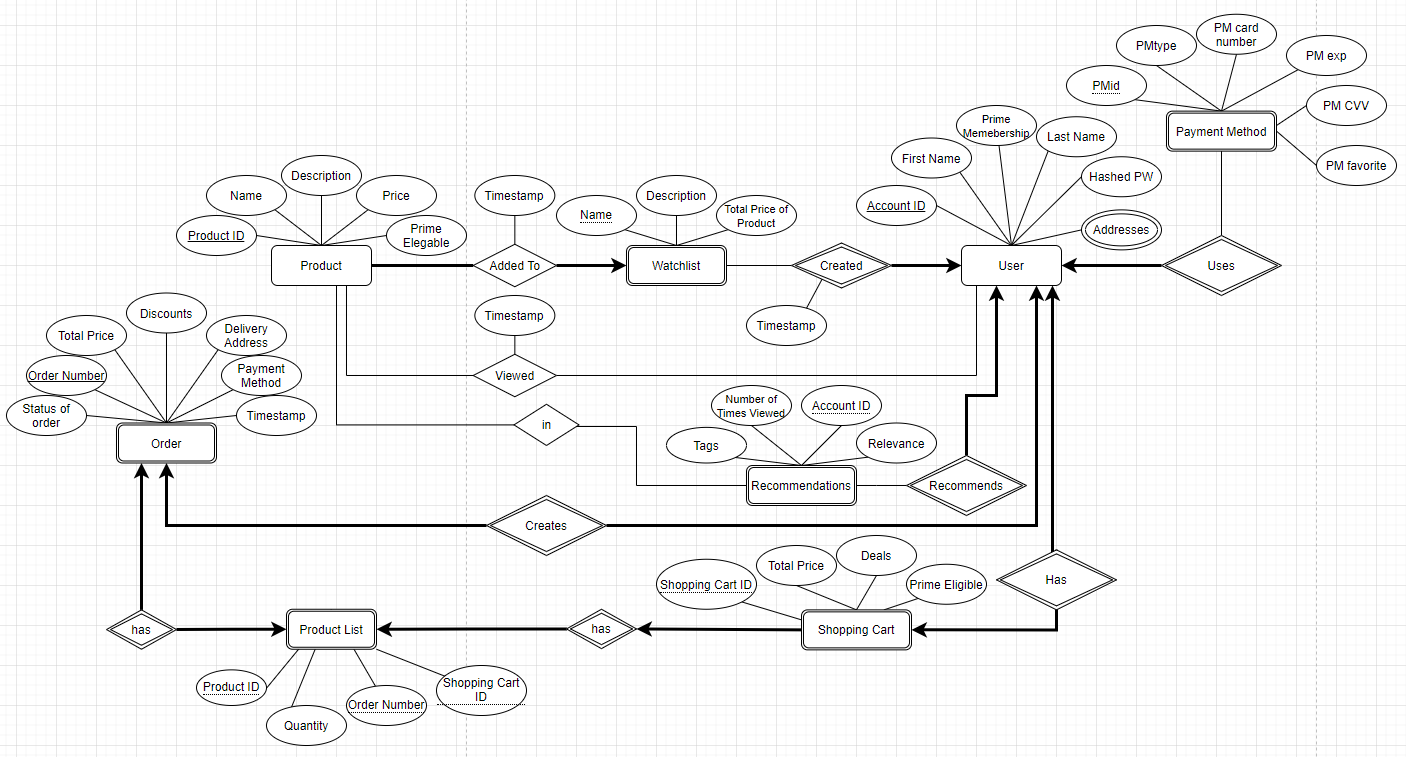
TimesViewed INTEGER NOT NULL,

LastViewed TIMESTAMP NOT NULL

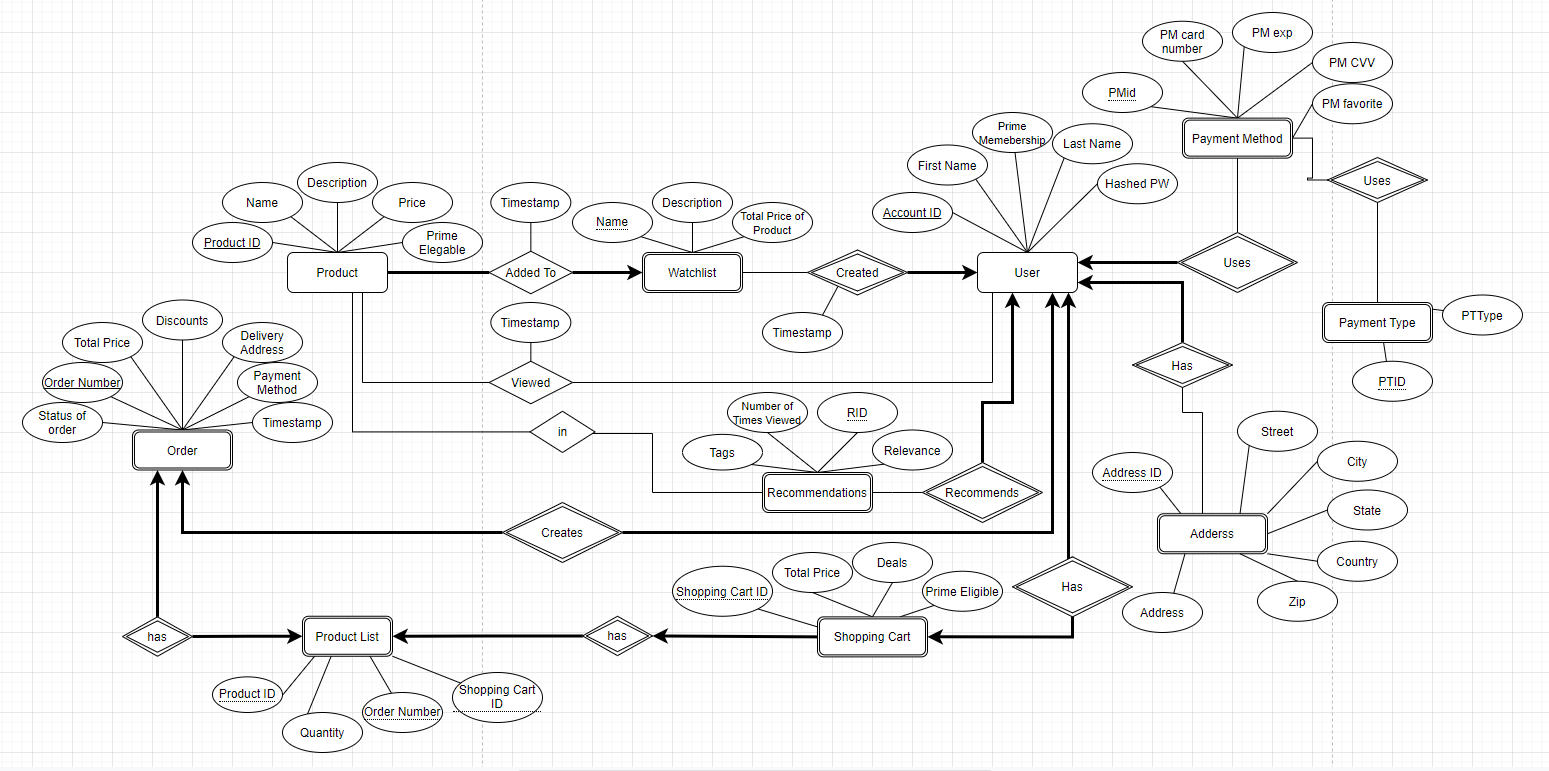
);

The Viewed table holds the information of products that a user has viewed. The PID references the Product table and the AccountID references the Account table. The Timesviewed increments each time the user views a product and the LastViewed changes to the last time the user views a product.

Old ER diagram:



New ER Diagram:



Relational Schema:

