CREATE TABLE Buyer (UserId int NOT NULL PRIMARY KEY, Password varchar(16), DateOfRegistration date, Email varchar(30), LastLogin date);

CREATE TABLE Seller (SellerId int NOT NULL PRIMARY KEY, Password varchar(16), DateOfRegistration date, Email varchar(30), AccountNumber int, LastLogin date, SellerRating int);

CREATE TABLE Address (AddressId int NOT NULL PRIMARY KEY, UserId int, Street varchar(30), Apartment varchar(30), Zip int);

CREATE TABLE Zip (ZipCode int NOT NULL PRIMARY KEY, County varchar(30), City varchar(30), State varchar(30), Country varchar(30));

CREATE TABLE PhoneNos (PhoneNo int NOT NULL PRIMARY KEY, UserId int);

CREATE TABLE Product (ProductId int NOT NULL PRIMARY KEY, UserId int, SubCategoryId int, SpecificatoinId int, ProductName varchar(30), Description varchar(255), BasePrice int, HighestBid int, Dimension varchar(30), Weight int, Manufacturer varchar(30), CountryOfOrigin varchar(30), PCondition int, ShippingOption int, QuantityAvailable int, LisitingEndDate Date, RefundReplacePolicy int, Discount int);

CREATE TABLE Keywords (KeywordId int NOT NULL PRIMARY KEY, ProductId int, Keyword varchar(30));

CREATE TABLE ProductImage (PImageId int NOT NULL PRIMARY KEY, ProductId int, ImageURL varchar(50));

CREATE TABLE WatchList (ProductId int, UserId int);

CREATE TABLE PurchaseHistory (PurchaseHistoryId int NOT NULL PRIMARY KEY, ProductId int, UserId int, BuyingPrice int, Quantity int);

CREATE TABLE Card (CardNumber int NOT NULL PRIMARY KEY, UserId int, CardType varchar(30), isDefault int, ExpiryDate date, CVV int, NameOnCard varchar(30), BillingAddress varchar(255));

CREATE TABLE SavedSearch (SavedSearchId int NOT NULL PRIMARY KEY, UserId int, SearchTerm varchar(30));

CREATE TABLE SavedSeller (UserId int, SellerId int);

CREATE TABLE ProductRating (UserId int, ProductId int, Rating int, Review varchar(255), DateOfReview date);

CREATE TABLE RecentlyViewed (UserId int, ProductId int, DateOfView date);

CREATE TABLE Category (CategoryId int NOT NULL PRIMARY KEY, CategoryName varchar(30));

CREATE TABLE SubCategory (SCategoryId int NOT NULL PRIMARY KEY, CategoryId int, SubCategoryName varchar(30));

CREATE TABLE Coupon (CouponId int NOT NULL PRIMARY KEY, ProductId int, CouponCode varchar(10), Description varchar(255), Validity date);

CREATE TABLE Specification (SpecificationId int NOT NULL PRIMARY KEY, SCategoryId int, SpecificationName varchar(255), SpecificationValue varchar(255));

CREATE TABLE Tracking (TrackingNumber int NOT NULL PRIMARY KEY, ProductId int, AddressId int, UserId int, StatusOfProduct int, ShipmentProvider varchar(30), EstimateDelivery date, ActualDelivery date);

CREATE TABLE Invoice (InvoiceId int NOT NULL PRIMARY KEY, CouponId int, UserId int, ProductId int, TaxId int, Price int, Discount int, Quantity int, InvoiceDate date);

CREATE TABLE Tax (TaxId int NOT NULL PRIMARY KEY, TaxRate int, State varchar(30));

CREATE TABLE Bid (UserId int, ProductId int, BidPrice int, BidDate Date);

CREATE TABLE ReturnP (ReturnId int NOT NULL PRIMARY KEY, UserId int, ProductId int, InvoiceId int, ReturnDate Date, NoOfDaysFromPurchase int, EligibleFlag int );

ALTER TABLE `Address` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `PhoneNos` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `Product` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `Product` ADD FOREIGN KEY (`SubCategoryId` ) REFERENCES `SubCategory` (`SCategoryId`) ON DELETE CASCADE;

ALTER TABLE `Product` ADD FOREIGN KEY (`SpecificationId`) REFERENCES `Specification`(`SpecificationId`) ON DELETE CASCADE;

ALTER TABLE `Keywords` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `ProductImage` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `WatchList` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `WatchList` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `PurchaseHistory` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `PurchaseHistory` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `Card` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `SavedSearch` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `SavedSeller` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `ProductRating` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `ProductRating` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `RecentlyViewed` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `RecentlyViewed` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `Subcategory` ADD FOREIGN KEY (`CategoryId`) REFERENCES `Category` (`CategoryId`) ON DELETE CASCADE;

ALTER TABLE `Coupon` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `Specification` ADD FOREIGN KEY (`SCategoryId`) REFERENCES `SubCategory` (`SCategoryId`) ON DELETE CASCADE;

ALTER TABLE `Tracking` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `Tracking` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `Tracking` ADD FOREIGN KEY (`AddressId`) REFERENCES `Address` (`AddressId`) ON DELETE CASCADE;

ALTER TABLE `Invoice` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `Invoice` ADD FOREIGN KEY (`CouponId`) REFERENCES `Coupon` (`CouponId`) ON DELETE CASCADE;

ALTER TABLE `Invoice` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `Invoice` ADD FOREIGN KEY (`TaxId`) REFERENCES `Tax` (`TaxId`) ON DELETE CASCADE;

ALTER TABLE `Bid` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `Bid` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

ALTER TABLE `Return` ADD FOREIGN KEY (`UserId`) REFERENCES `BuyerSeller` (`UserId`) ON DELETE CASCADE;

ALTER TABLE `Return` ADD FOREIGN KEY (`ProductId`) REFERENCES `Product` (`ProductId`) ON DELETE CASCADE;

DROP TABLE Address;

DROP TABLE Zip;

DROP TABLE PhoneNos;

DROP TABLE Keywords;

DROP TABLE ProductImage;

DROP TABLE WatchList;

DROP TABLE PurchaseHistory;

DROP TABLE Card;

DROP TABLE SavedSearch;

DROP TABLE SavedSeller;

DROP TABLE ProductRating;

DROP TABLE RecentlyViewed;

DROP TABLE Category;

DROP TABLE Subcategory;

DROP TABLE Coupon;

DROP TABLE Specification;

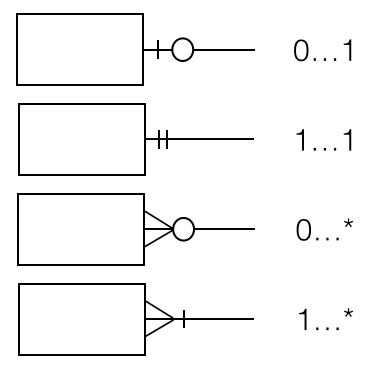
DROP TABLE Tracking;

DROP TABLE Invoice;

DROP TABLE Tax;

DROP TABLE BuyerSeller;

DROP TABLE Product;



Changes:

Tracking: Invoice Id (FK) --Done

NoOfDaysFromPurchase

**Procedures**

1 To check whether product is eligible for return or not

CREATE OR REPLACE PROCEDURE CHECKRETURNELIGIBILE AS

CURSOR C1

is

SELECT \* FROM RETURNS;

BEGIN

FOR R

IN C1

LOOP

IF X.NoOfDaysFromPurchase > 60 THEN

update RETURNS T

set EligibleFlag = 0

WHERE R.ReturnId=T.ReturnId;

END IF;

END LOOP;

END;

2 To make product quantity 0 once the the listing end date is reached as it will be no more for sale

CREATE OR REPLACE PROCEDURE AdStatus AS

CURSOR C1

is

SELECT \* FROM Product;

BEGIN

FOR P

IN C1

LOOP

IF SYSDATE > P.LisitingEndDate THEN

update Product P1

set Quantity = 0

where P1.ADID= P.ADID;

END IF;

END LOOP;

END;

**Triggers**

1 To update the seller rating

CREATE OR REPLACE TRIGGER updateSellerRatings AFTER

INSERT OR UPDATE OF rating ON review

FOR EACH ROW

DECLARE

new\_rating NUMBER(2, 1);

seller\_id\_to\_update VARCHAR(255);

BEGIN

new\_rating := :new.rating;

SELECT

seller\_id

INTO seller\_id\_to\_update

FROM product WHERE product\_id = :new.product\_id;

UPDATE seller SET average\_rating = ( ( average\_rating \* rating\_count ) + new\_rating ) / ( rating\_count + 1 ),

rating\_count = rating\_count + 1 WHERE seller\_id = seller\_id\_to\_update;

END;

2 To set default card for the buyer

CREATE OR REPLACE PROCEDURE setDefaultCard (

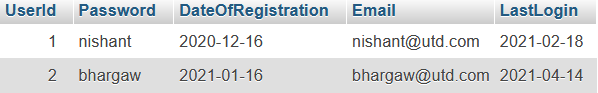
cardnumber IN INTEGER,

userid\_up IN VARCHAR)

AS BEGIN UPDATE card\_info SET is\_default = 1 WHERE UserId = userid\_up AND CardNumber = cardnumber;

END setDefaultCard

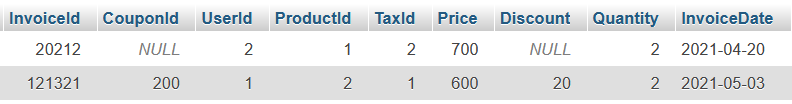
Buyer table :



Seller table:



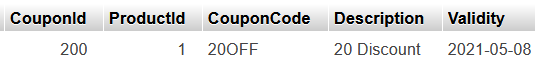
Invoice:



Address



Coupon



Keywords :

