A simple database supporting an online book seller

Tables about Books and Authors

```
CREATE TABLE Book (
   Isbn                  INTEGER,
   Title                 CHAR[120] NOT NULL,
   Synopsis              CHAR[500],
   ListPrice             CURRENCY NOT NULL,
   AmazonPrice           CURRENCY NOT NULL,
   SavingsInPrice        CURRENCY NOT NULL,   /* redundant
   AveShipLag            INTEGER,
   AveCustRating         REAL,
   SalesRank             INTEGER,
   CoverArt              FILE,
   Format                CHAR[4] NOT NULL,
   CopiesInStock         INTEGER,
   PublisherName         CHAR[120] NOT NULL,  /*Remove NOT NULL if you want 0 or 1
   PublicationDate       DATE NOT NULL,
   PublisherComment  CHAR[500],
   PublicationCommentDate DATE,
   PRIMARY KEY (Isbn),
   FOREIGN KEY (PublisherName) REFERENCES Publisher,
        ON DELETE NO ACTION,  ON UPDATE CASCADE,
   CHECK (Format = ‘hard’ OR Format = ‘soft’ OR Format = ‘audi’
                                  OR Format = ‘cd’ OR Format = ‘digital’)
        /* alternatively, CHECK (Format IN (‘hard’, ‘soft’, ‘audi’, ‘cd’, ‘digital’))
   CHECK (AmazonPrice + SavingsInPrice = ListPrice)
)

CREATE TABLE Author (
   AuthorName      CHAR[120],
   AuthorBirthDate DATE,
   AuthorAddress   ADDRESS,
   AuthorBiography  FILE,
   PRIMARY KEY (AuthorName, AuthorBirthDate)
)

CREATE TABLE WrittenBy (/*Books are written by authors
   Isbn INTEGER,
   AuthorName CHAR[120],
   AuthorBirthDate DATE,
   OrderOfAuthorship INTEGER NOT NULL,
   AuthorComment  FILE,
   AuthorCommentDate DATE,
   PRIMARY KEY (Isbn, AuthorName, AuthorBirthDate),
   FOREIGN KEY (Isbn) REFERENCES Book,
      ON DELETE CASCADE, ON UPDATE CASCADE,
   FOREIGN KEY (AuthorName, AuthorBirthDate) REFERENCES Author,
      ON DELETE CASCADE, ON UPDATE CASCADE)
```

```
CREATE TABLE Publisher (
   PublisherName      CHAR[120],
   PublisherAddress   ADDRESS,
   PRIMARY KEY (PublisherName)
)

// insure participation constraint of Publisher in Book (you were asked to write this)
CREATE ASSERTION PublisherBookConstraint
CHECK (NOT EXISTS (SELECT *
                   FROM Publisher P
                   WHERE P.PublisherName
                                    NOT IN (SELECT B.PublisherName
                                            FROM Book B)))


// insure participation constraint of Books in WrittenBy
CREATE ASSERTION BookWrittenByConstraint
CHECK (NOT EXISTS
                  (SELECT *
                   FROM Book B
                   WHERE B.Isbn NOT IN (SELECT W.Isbn FROM WrittenBy W)))


// insure participation constraint of Authors in WrittenBy
CREATE ASSERTION AuthorWrittenConstraint
CHECK (NOT EXISTS
                  (SELECT *
                   FROM Author A
                   WHERE A.AuthorName, A.AuthorBirthDate
                            NOT IN (SELECT W.AuthorName, W.AuthorBirthDate
                                    FROM WrittenBy W)))
```

Tables about Customers and Customer Service

```
CREATE TABLE Customer (/* Customers identified by email address
   CustEmailAddr    CHAR[120],
   CustName    CHAR[120] NOT NULL,
   CustPassword  CHAR[20] NOT NULL,
   PRIMARY KEY (CustEmailAddr)
)
```

```
//Customers can request notification about new books by an author
CREATE TABLE AlertTo (
   CustEmailAddress  CHAR[120],
   DateOfAlertRequest  DATE NOT NULL,
   AuthorName  CHAR[120],
   AuthorBirthDate DATE,
   PRIMARY KEY (UserEmailAddr, AuthorName, AuthorBirthDate),
   FOREIGN KEY (AuthorName, AuthorBirthDate) REFERENCES Author
     ON DELETE CASCADE, ON UPDATE CASCADE,
   FOREIGN KEY (CustEmailAddr) REFERENCES Customer
      ON DELETE NO ACTION, ON UPDATE CASCADE
)

CREATE TABLE Account (/* Customers can have zero or more accounts
   CustEmailAddr    CHAR[120],
   CreditCardNumber  INTEGER,
   ShippingAddr   ADDRESS NOT NULL,
   DateOpened    DATE NOT NULL,
   PRIMARY KEY (CustEmailAddr, CreditCardNumber),
   FOREIGN KEY (CustEmailAddr) REFERENCES Customer
        ON DELETE CASCADE, ON UPDATE CASCADE
)


// insure participation constraint of Customer in /Accounts – every customer have at least
// one account
CREATE ASSERTION CustomerAccountConstraint
CHECK (NOT EXISTS (SELECT *
                              FROM Customer C
                              WHERE C.CustEmailAddr
                                   NOT IN (SELECT A.CustEmailAddr FROM Account A)))
```

Tables about Purchases and Shipments

```
//Transaction (purchases) are made on a customer account
CREATE TABLE Transaction (
  TransNumber   INTEGER,
  OrderDate   DATE,
  PaymentClearanceDate  DATE,       /* if NULL, then payment has not cleared */
  CustEmailAddr    CHAR[120] NOT NULL,
  CreditCardNo   INTEGER NOT NULL,
  PRIMARY KEY (TransNum),
  FOREIGN KEY (CustEmailAddr, CreditCardNo) REFERENCES Account
     ON DELETE NO ACTION, ON UPDATE CASCADE
 )
```

```
CREATE TABLE Shipment (/* A record of purchases awaiting or when shipment
   ShipId INTEGER,
   ShipCost CURRENCY,
   ShipDate DATE,                    /* if this is NULL, then not shipped yet */
   TransNumber INTEGER NOT NULL,
   PRIMARY KEY (ShipId),
   FOREIGN KEY (TransNumber) REFERENCES Transaction
      ON DELETE CASCADE, ON UPDATE CASCADE
)

// insure participation constraint on Transaction in Shipment (at least one shipment
// per transaction
CREATE ASSERTION TransactionsShipmentConstraint
CHECK (NOT EXISTS (SELECT *
                   FROM Transaction T
                   WHERE T.TransNumber NOT IN (SELECT S.TransNumber
                                               FROM Shipment S)))


CREATE TABLE BookShipment (/* A quantity of book associated with a shipment and
                           /* therefore transaction
    Quantity        INTEGER,
    ShipId          INTEGER,
    Isbn            INTEGER,
    PRIMARY KEY (ShipId, Isbn),
    FOREIGN KEY (ShipId) REFERENCES Shipment
      ON DELETE CASCADE, ON UPDATE CASCADE,
    FOREIGN KEY (Isbn) REFERENCES Book
      ON DELETE NO ACTION, ON UPDATE CASCADE
)
```