**Class Exercise – 10.1**

**Name: Ashlyn Cooper**

**Clemson/Assignment ID: CONCUR**

**Submission**: Save this Word document with your answers as a PDF file and upload the PDF file to Canvas.

**Transactions**

**ABC Markets sell products to customers. The relational diagram shown in Figure P10.6 represents the main entities for ABC’s database. Note the following important characteristics:**

* **A customer may make many purchases, each one represented by an invoice.** 
  + **The CUS\_BALANCE is updated with each credit purchase or payment and represents the amount the customer owes.**
  + **The CUS\_BALANCE is increased (+) with every credit purchase and decreased (-) with every customer payment.**
  + **The date of last purchase is updated with each new purchase made by the customer.**
  + **The date of last payment is updated with each new payment made by the customer.**
* **An invoice represents a product purchase by a customer.** 
  + **An INVOICE can have many invoice LINEs, one for each product purchased.**
  + **The INV\_TOTAL represents the total cost of invoice including taxes.**
  + **The INV\_TERMS can be “30,” “60,” or “90” (representing the number of days of credit) or “CASH,” “CHECK,” or “CC.”**
  + **The invoice status can be “OPEN,” “PAID,” or “CANCEL.”**
* **A product’s quantity on hand (P\_QTYOH) is updated (decreased) with each product sale.**
* **A customer may make many payments. The payment type (PMT\_TYPE) can be one of the following:**
  + **“CASH” for cash payments.**
  + **“CHECK” for check payments**
  + **“CC” for credit card payments**
* **The payment details (PMT\_DETAILS) are used to record data about check or credit card payments:**
  + **The bank, account number, and check number for check payments**
  + **The issuer, credit card number, and expiration date for credit card payments.**

**A picture containing text, screenshot, sign

Description automatically generated**

**Using this database, write the SQL code to represent each one of the following transactions. Use BEGIN TRANSACTION and COMMIT to group the SQL statements in logical transactions. [Note: Not all entities and attributes are represented in this example. Use only the attributes indicated.]**

1. **On May 11, 2016, customer ‘10010’ makes a credit purchase (30 days) of one unit of product ‘11QER/31’ with a unit price of $110.00; the tax rate is 8 percent. The invoice number is 10983, and this invoice has only one product line.**

BEGIN TRANSACTION

INSERT INTO INVOICE

VALUES(10983, ‘10010’, ’11-May-2018’, 118.80,’30’, ‘OPEN’);

INSERT INTO LINE

VALUES(10983, 1, ‘11QER/31’, 1, 110.00);

UPDATE PRODUCT

SET P\_OTYOH= P\_QTYOH-1

WHERE P\_CODE = ‘11QER/31’;

UPDATE CUSTOMER

SET CUS\_DATELSTPUR = ’11-May-2018’

CUS\_BALANCE=CUS\_BALANCE+118.80

WHERE CUS\_CODE = ‘10010’;

COMMIT

1. **On June 3, 2016, customer ‘10010’ makes a payment of $100 in cash. The payment ID is 3428.**

BEGIN TRANSACTION

INSERT INTO PAYMENTS

VALUES(3428,’03-Jun-2016’,‘10010’,100.00,‘CASH’, ‘None’);

UPDATE CUSTOMER

SET CUS\_DATELSTPMT = ’03-Jun-2016’,

CUS\_BALANCE= CUS\_BALANCE – 100.00

WHERE CUS\_CODE = ‘10010’;

COMMIT