National University of Computer & Emerging Sciences, Karachi

Fall-2023 School of Computing (BSCS, BSSE, BSCY, BSAI)

Assignment # 04

Subject: Database Systems -CS2005	**Post Date: 21/11/2023**
Total Marks: 26	**Due Date:** 30/11/2023
Course Instructors: Dr. Zulfiqar Memon, Dr. Anam Qureshi, Ms. Romasha Khurshid, Mr. Shahroz Bakht, Ms. Fizza Aqeel, Ms. Fizza Mansoor, Mr. Omar Qureshi, Ms. Abeer Gauher, Mr. Muhammad Ali Fatmi	

Instructions to be strictly followed.

- For all questions submit a doc/pdf file**.**
- It should be obvious that submitting your work after the due date will result in zero points being awarded.
- Plagiarism (copying/cheating) and late submissions result in a zero mark.

Marks: 2*3 = 06

1. Suppose that you are a manufacturer of product ABC, which is composed of parts A, B, and C. Each time a new product ABC is created, it must be added to the product inventory, using the PROD_QOH in a table named PRODUCT. Also, each time the product is created, the parts inventory, using PART_QOH in a table named PART, must be reduced by one each of parts A, B, and C. The sample database contents are shown in Table P10.1.

TABLE P10.1

TABLE NAME: PRODUCT

PROD_CODE	PROD_QOH
ABC	1,205

TABLE NAME: PART

PART_CODE	PART_QOH
A	567
B	98
C	549

Given the preceding information, answer Questions a through e.

a. How many database requests can you identify for an inventory update for both PRODUCT and PART?
b. Using SQL, write each database request you identified in Step a.
c. Write the complete transaction(s).

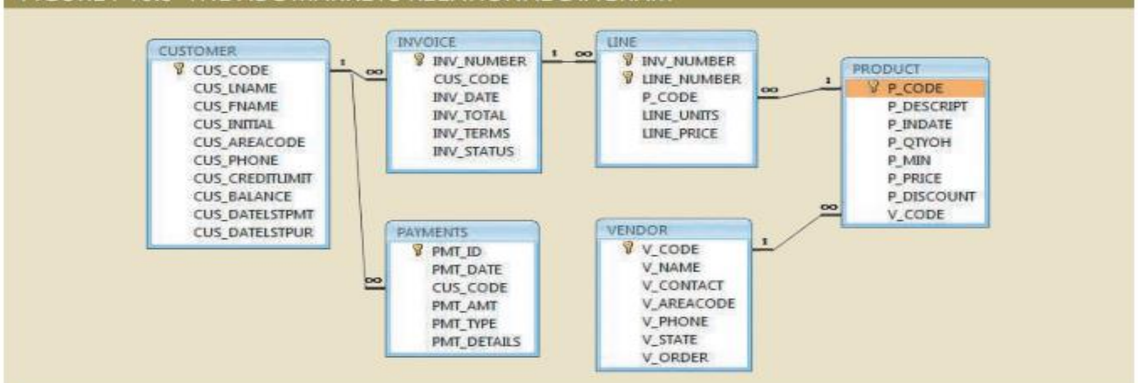
Marks: 2 * 2= 04

2. 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 the 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.

Note: Not all entities and attributes are represented in this example. Use only the attributes indicated.

Using this database, write the SQL code to represent each of the following transactions. Use BEGIN TRANSACTION and COMMIT to group the SQL statements in logical transactions.

FIGURE P10.6 THE ABC MARKETS RELATIONAL DIAGRAM



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

Marks: 4 * 4= 16

3. Which of the following schedules is (conflict) serializable? For each serializable schedule, determine the equivalent serial schedules.
   a. $r_1(X); r_3(X); w_1(X); r_2(X); w_3(X);$
   b. $r_1(X); r_3(X); w_3(X); w_1(X); r_2(X);$
   c. $r_3(X); r_2(X); w_3(X); r_1(X); w_1(X);$
   d. $r_3(X); r_2(X); r_1(X); w_3(X); w_1(X);$
4. Consider the three transactions T_1, T_2, and T_3, and the schedules S_1 and S_2 given below. Draw the serializability (precedence) graphs for S_1 and S_2, and state whether each schedule is serializable or not. If a schedule is serializable, write down the equivalent serial schedule(s).

   T_1: $r_1 (X); r_1 (Z); w_1 (X);$

   T_2: $r_2 (Z); r_2 (Y); w_2 (Z); w_2 (Y);$

   T_3: $r_3 (X); r_3 (Y); w_3 (Y);$

   S_1: $r_1 (X); r_2 (Z); r_1 (Z); r_3 (X); r_3 (Y); w_1 (X); w_3 (Y); r_2 (Y); w_2 (Z); w_2 (Y);$

   S_2: $r_1 (X); r_2 (Z); r_3 (X); r_1 (Z); r_2 (Y); r_3 (Y); w_1 (X); w_2 (Z); w_3 (Y); w_2 (Y);$
5. Consider schedules S_3, S_4, and S_5 below. Determine whether each schedule is strict, cascadeless, recoverable, or nonrecoverable. (Determine the strictest recoverability condition that each schedule satisfies.)

   S_3: $r_1 (X); r_2 (Z); r_1 (Z); r_3 (X); r_3 (Y); w_1 (X); c_1; w_3 (Y); c_3; r_2 (Y); w_2 (Z); w_2 (Y); c_2;$

   S_4: $r_1 (X); r_2 (Z); r_1 (Z); r_3 (X); r_3 (Y); w_1 (X); w_3 (Y); r_2 (Y); w_2 (Z); w_2 (Y); c_1; c_2; c_3;$

   S_5: $r_1 (X); r_2 (Z); r_3 (X); r_1 (Z); r_2 (Y); r_3 (Y); w_1 (X); c_1; w_2 (Z); w_3 (Y); w_2 (Y); c_3; c_2;$
6. Modify the data structures for multiple-mode locks and the algorithms for read_lock(X), write_lock(X), and unlock(X) so that upgrading and downgrading of locks are possible. (Hint: The lock needs to check the transaction id(s) that hold the lock, if any.)