**Assignment 11**

**Swapnil Shashikant Kamate**

**Part1**

Problems 31-34

31)Create a trigger named trg\_line\_total to write the LINE\_TOTAL value in the LINE table every time you add a new LINE row. (The LINE\_TOTAL value is the product of the LINE\_UNITS and LINE\_PRICE values.)

**Answer:**

CREATE OR REPLACE TRIGGER TRG\_LINE\_TOTAL

BEFORE INSERT ON LINE

FOR EACH ROW

INSERT INTO LINE

SET ACTION = 'insert',

LINE.LINE\_TOTAL = LINE\_UNITS\*LINE\_PRICE;

**Table

Description automatically generated**

32)Create a trigger named trg\_line\_prod that automatically updates the quantity on hand for each product sold after a new LINE row is added.

**Answer:**

DELIMITER $$

CREATE TRIGGER TRG\_LINE\_PROD

AFTER UPDATE ON PRODUCT

FOR EACH ROW

BEGIN

UPDATE PRODUCT

SET P\_QOH = P\_QOH-NEW.LINE\_UNITS

WHERE PRODUCT.P\_CODE = NEW.P\_CODE;

END$$

DELIMITER;

**Graphical user interface

Description automatically generated with low confidence**

33)Create a stored procedure named prc\_inv\_amounts to update the INV\_SUBTOTAL, INV\_TAX, and INV\_TOTAL. The procedure takes the invoice number as a parameter. The INV\_SUBTOTAL is the sum of the LINE\_TOTAL amounts for the invoice, the INV\_TAX is the product of the INV\_SUBTOTAL and the tax rate (8 percent), and the INV\_TOTAL is the sum of the INV\_SUBTOTAL and the INV\_TAX.

**Answer:**

DELIMITER //

CREATE PROCEDURE PRC\_INV\_AMOUNTS(IN INV\_NUMBER INT)

BEGIN

UPDATE INVOICE

SET INV\_SUBTOTAL = SUM(LINE.LINE\_TOTAL);

UPDATE INVOICE

SET INV\_TAX = INV\_SUBTOTAL\*0.08;

UPDATE INVOICE

SET INV\_TOTAL = INV\_SUBTOTAL + INV\_TAX;

END //

**A picture containing text, scoreboard

Description automatically generated**

**Part-2**

**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. 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?

**Answer:**

For an inventory update for both PRODUCT and PART we need 2 database requests.

b**.** Using SQL, write each database request you identified in Step a.

**Answer:**

UPDATE PRODUCT

SET PROD\_QOH = PROD\_QOH + 1

WHERE PROD\_CODE = ABC

UPDATE PART

SET PROD\_QOH = PROD\_QOH – 1

WHERE PART\_CODE = A OR PART\_CODE = B OR PART\_CODE = C

c. Write the complete transaction(s).

**Answer:**

START TRANSACTION

UPDATE PRODUCT

SET PROD\_QOH = PROD\_QOH + 1

WHERE PROD\_CODE = ABC

UPDATE PART

SET PROD\_QOH = PROD\_QOH – 1

WHERE PART\_CODE = A OR PART\_CODE = B OR PART\_CODE = C

COMMIT;

d. Write the transaction log, using Table 10.1 as your template.

**Answer:**

With PROD\_CODE as ABC and PROD\_QOH as 1205, the values where PART\_ID is A,B and C is 567,98 and 549

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **NUM** | **OPERATION** | **TABLE** | **ROW\_ID** | **ATT** | **BEFORE** | **AFTER** |
| 1 | 1A3 | START | START |  |  |  |  |
| 2 | 1A3 | UPDATE | PRODUCT | ABC | PROD\_QOH | 1205 | 1206 |
| 3 | 1A3 | UPDATE | PART | A | PART\_QOH | 567 | 566 |
| 4 | 1A3 | UPDATE | PART | B | PART\_QOH | 98 | 97 |
| 5 | 1A3 | UPDATE | PART | C | PART\_QOH | 549 | 548 |
| 6 | 1A3 | COMMIT | END |  |  |  |  |

e. Using the transaction log you created in Step d, trace its use in database recovery.

**Answer:**

For the database recovery we actually go from last id to first. In my case I will start with ID =6 and go backwards. We identify the next step to reverse and move from last transaction back to first.

2) Describe the three most common problems with concurrent transaction execution. Explain how concurrency control can be used to avoid those problems.

**Answer:**

The 3 most common concurrent transaction execution problems are lost updates, uncommitted data and inconsistent retrievals.

Concurrency control can help avoid these problems through its ability to coordinate simultaneous executions of transactions in a multiprocessing database system. This can be achieved with various methods like use of schedulers and locks.

3) What DBMS component is responsible for concurrency control? How is this feature used to resolve conflicts?

**Answer:**

Scheduler is responsible for concurrency control in DBMS. The main advantage of using scheduler is that you can perform many operations in DBMS or in these many operations each of these transactions can be executed with the use of scheduler.

This is how it can be used to resolve conflicts.