CIS 229

Date: 11/27/18

Cedric L Mulumba

Prof: Bob Desilets

Chapter 9

**Hands-On Assignments Part I**

|  |
| --- |
| **Assignment 9-1: Creating a Trigger to Handle Product Restocking** |
| CREATE OR REPLACE TRIGGER bb\_reorder\_trg  AFTER UPDATE OF stock ON bb\_product  FOR EACH ROW  DECLARE  v\_onorder\_num NUMBER(4);  BEGIN  If :NEW.stock <= :NEW.reorder THEN  SELECT SUM(qty)  INTO v\_onorder\_num  FROM bb\_product\_request  WHERE idProduct = :NEW.idProduct  AND dtRecd IS NULL;  IF v\_onorder\_num IS NULL THEN v\_onorder\_num := 0; END IF;  IF v\_onorder\_num = 0 THEN  INSERT INTO bb\_product\_request (idRequest, idProduct, dtRequest, qty)  VALUES (bb\_prodreq\_seq.NEXTVAL, :NEW.idProduct, SYSDATE, :NEW.reorder);  END IF;  END IF;  END; |
|  |
| SELECT stock, reorder  FROM bb\_product  WHERE idProduct = 4; |
|  |
| UPDATE bb\_product  SET stock = 25  WHERE idProduct = 4; |
|  |
| SELECT \*  FROM bb\_product\_request; |
|  |
| rollback; |
|  |
| ALTER TRIGGER bb\_reorder\_trg DISABLE; |
|  |
| **Assignment 9-2: Updating Stock Information When a Product Request Is Filled** |
| insert into bb\_product\_request  (idrequest, idproduct, dtrequest, qty)  values (3, 5, sysdate, 45); |
|  |
| create or replace trigger bb\_reqfill\_trg  after update of dtrecd on bb\_product\_request  for each row    begin    update bb\_product  set stock = :new.qty + stock  where idproduct = :new.idproduct;  end; |
|  |
| select stock, reorder  from bb\_product  where idproduct = 5; |
|  |
| select \*  from bb\_product\_request  where idproduct = 5; |
|  |
| update bb\_product\_request  set dtrecd = sysdate, cost = 225  where idproduct = 5; |
|  |
| select stock, reorder  from bb\_product  where idproduct = 5; |
|  |
| select \*  from bb\_product\_request  where idproduct = 5; |
|  |
| rollback; |
|  |
| **Assignment 9-3: Updating the Stock Level If a Product Fulfillment Is Canceled** |
| CREATE OR REPLACE TRIGGER BB\_REQFILL\_TRG  AFTER UPDATE OF  DTRECD ON Bb\_PRODUCT\_REQUEST  FOR EACH ROW  BEGIN  UPDATE BB\_PRODUCT  SET STOCK = STOCK + :new.QTY  WHERE idproduct = :new.idproduct;  END; |
|  |
| INSERT INTO bb\_product\_request (idRequest, idProduct, dtRequest, qty,  dtRecd, cost)  VALUES (4, 5, SYSDATE, 45, '15-JUN-2012',225); |
|  |
| UPDATE bb\_product  SET stock = 86  WHERE idProduct = 5;  COMMIT; |
|  |
| UPDATE bb\_product\_request  SET dtRecd = NULL  WHERE idRequest = 4; |
|  |
| **Assignment 9-4: Updating Stock Levels When an Order Is Canceled** |
| CREATE OR REPLACE TRIGGER bb\_ordcancel\_trg  AFTER INSERT ON bb\_basketstatus  FOR EACH ROW  WHEN (NEW.idstage = 4)  DECLARE CURSOR mack\_trg\_cursor  IS  SELECT i.quantity, i.idbasket, i.idproduct  FROM bb\_basketitem i;  BEGIN  FOR rec\_cur IN mack\_trg\_cursor  LOOP  UPDATE bb\_product  SET stock = stock + rec\_cur.quantity  WHERE rec\_cur.idproduct = idproduct;  UPDATE bb\_basket  SET orderplaced = 0  WHERE rec\_cur.idbasket = :NEW.idbasket;  END LOOP;  END |
|  |
| INSERT INTO bb\_basketstatus (idStatus, idBasket, idStage, dtStage)  VALUES (bb\_status\_seq.NEXTVAL, 6, 4, SYSDATE); |
|  |
| select \*  from bb\_basketstatus  where idBasket = 6  and idStage = 4; |
|  |
| ALTER TRIGGER bb\_ordcancel\_trg DISABLE; |
|  |
| **Assignment 9-8: Maintaining an Audit Trail of Product Table Changes** |
| ALTER TRIGGER sales\_date\_trg DISABLE; |
|  |
| UPDATE bb\_product  SET salestart = '05-MAY-2012',  saleend = '12-MAY-2012',  saleprice = 9  WHERE idProduct = 10; |
|  |
| CREATE TABLE bb\_prodchg\_audit  (user\_id VARCHAR2(10),  chg\_date DATE,  name\_old VARCHAR2(25),  name\_new VARCHAR2(25),  price\_old NUMBER(5,2),  price\_new NUMBER(5,2),  start\_old DATE,  start\_new DATE,  end\_old DATE,  end\_new DATE,  sale\_old NUMBER(5,2),  sale\_new NUMBER(5,2) ); |
|  |
| select \*  from bb\_prodchg\_audit; |
|  |