Assignment 9

CS1103

Student Name: Omar Sebri

Student ID: Omar Sebri

Part A:

Code:

```
DELIMITER $$
CREATE procedure adjustPrice(itemID INT)
BEGIN
if ( EXISTS(SELECT item_id FROM menu_items WHERE item_id=itemID)) then
      if((SELECT item_price FROM menu_items WHERE item_id=itemID)<2) then
            UPDATE menu_items
                  SET item_price = (item_price*102)/100
                  WHERE item_id=itemID;
      ELSEIF ((SELECT item_price FROM menu_items WHERE item_id=itemID)<4)
then
            UPDATE menu_items
                  SET item_price = (item_price*104)/100
                  WHERE item_id=itemID;
      ELSE
            UPDATE menu_items
                  SET item_price = (item_price*105)/100
                  WHERE item_id=itemID;
      END if ;
ELSE SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT ='item not found' ;
END if;
END$$
```

Testing:

Calling Procedure:

```
call adjustPrice(1);
call adjustPrice(6);
call adjustPrice(7);
```

Before:

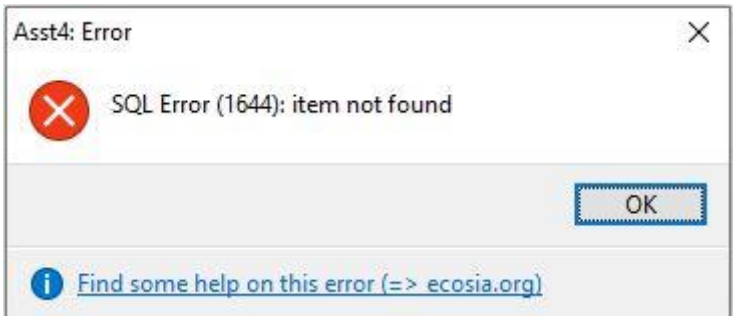
Item_ID	Item_name	Item_price	Item_category
1	Coffee (Original Blend, Dark Roast or Decaf)Small	1,59	11
6	Iced Coffee (Original or Dark Roast)Large	2,90	12
7	California Turkey	6,49	13

After:

Item_ID	Item_name	Item_price	Item_category
1	Coffee (Original Blend, Dark Roast or Decaf)Small	1,62	11
6	Iced Coffee (Original or Dark Roast)Large	3,02	12
7	California Turkey	6,81	13

Using a non-existent Item ID:

```
call adjustPrice(19);
```



Part B:

Code:

```
DELIMITER $$
CREATE PROCEDURE multisale(itemID INT,quantity INT,saletype INT,storeID
INT,employeeID INT)
BEGIN
declare START INT DEFAULT 0;
if(EXISTS(SELECT item_id FROM menu_items WHERE item_id=itemID)
AND EXISTS(SELECT sale_type_id FROM sales_type WHERE sale_type_id=saletype)
AND EXISTS(SELECT employee_id FROM employees WHERE employee_id=employeeid)
AND EXISTS(SELECT shop_id FROM tim_store WHERE shop_id=storeID)) then
Select
@lastint:=MAX(sale_id)
FROM sales_record;
INSERT INTO sales_record
VALUES(@lastint+1,CURDATE(),saletype,storeid,employeeid);
while(START < quantity) DO
      INSERT INTO line_items VALUES(START+1,(SELECT MAX(sale_id) FROM
sales_record),itemID);
      set START = START + 1;
END while ;
SELECT item_price*quantity AS TotalSalePrice FROM menu_items
      WHERE item_id=itemID;
ELSE SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT ='invalid input' ;
END if;
END$$
```

Testing:

```
SELECT * FROM sales_record
WHERE sale_id = (SELECT MAX(sale_id) FROM sales_record) ;
```

Sale_ID	Date_and_time	Sale_type	Store	Employee
10 021	2022-04-03 00:00:00	2	103	1 001

```
SELECT sale_id, line_item_id, item_name FROM
 line_items NATURAL JOIN menu_items
 WHERE  sale_id = (SELECT MAX(sale_id) FROM sales_record);
```

sale_id	line_item_id	item_name
10 021	1	California Turkey
10 021	2	California Turkey
10 021	3	California Turkey
10 021	4	California Turkey

PS: Since my menu doesn't have donuts I will instead use Turkey bacon club.

Calling procedures:

```
CALL multisale(1,3,2,103,1001);
```

TotalSalePrice
4,86

```
CALL multisale(8,4,2,103,1001);
```

line_items (33r × 3c)

Line_item_ID	Sale_ID	Item_ID
2	10 020	1
1	10 020	1
3	10 020	1
3	10 021	7
2	10 021	7
1	10 021	7
4	10 021	7

Sales_rcord table after calling the two procedures:

sales_record (21r × 5c)

Sale_ID	Date_and_time	Sale_type	Store	Employee
10 017	2022-04-03 00:00:00	2	103	1 001
10 018	2022-04-03 00:00:00	2	103	1 001
10 019	2022-04-03 00:00:00	2	103	1 001
10 020	2022-04-03 00:00:00	2	103	1 001
10 021	2022-04-03 00:00:00	2	103	1 001
10 022	2022-04-03 00:00:00	2	103	1 001
10 023	2022-04-03 00:00:00	2	103	1 001

Line_items Table after calling the two procedures:

Line_item_ID	Sale_ID	Item_ID
1	10 022	1
2	10 022	1
3	10 022	1
3	10 023	8
1	10 023	8
2	10 023	8
4	10 023	8

```
SELECT * FROM sales_record
WHERE sale_id = (SELECT MAX(sale_id) FROM sales_record) ;
```

Sale_ID	Date_and_time	Sale_type	Store	Employee
10 023	2022-04-03 00:00:00	2	103	1 001

```
SELECT sale_id, line_item_id, item_name FROM
 line_items NATURAL JOIN menu_items
 WHERE  sale_id = (SELECT MAX(sale_id) FROM sales_record);
```

sale_id	line_item_id	item_name
10 023	1	Turkey Bacon Club
10 023	2	Turkey Bacon Club
10 023	3	Turkey Bacon Club
10 023	4	Turkey Bacon Club

Testing Invalid Inputs:

Non-existent item id:

```
CALL multisale(15,3,2,103,1001);
```

Negative quantity:

```
CALL multisale(8,-1,2,103,1001);
```

Non-existent sale type:

```
CALL multisale(8,3,25,103,1001);
```

Non-existent store:

```
CALL multisale(8,3,2,133,1001);
```

Non-existent Employee:

```
CALL multisale(8,3,2,103,3);
```

All of the previous calls yield the same result: an error