DBMS Project Report : FlipMart

User Guide

Functionalities:

The program starts asking the user to login either as an administrator or a customer in the Retail Store System. If the user is an Admin, they are authenticated using a username and password. Both the customer and the admin will be able to access the records of the branch they are registered in. The interface provides various functionalities to the two types of users as listed below.

Admin can perform the following functions:

1. View the available products at their branch.
2. Add a product to their branch. It may or may not be available at other branches.
3. Alter the quantity of a product available at the branch warehouse. This may be used when a new stock is brought in or the products are sold from some other outlet as well.
4. Change the password they have to use for authentication.
5. Reward coupons to the customers either randomly or on the basis of the maximum number of orders.
6. Add a new customer to the branch.
7. View the sales performance and statistics of the branch.

Customer can perform the following functions:

1. View the available products at their branch.
2. View the contents of their current cart.
3. Add a product to their cart with the quantity they want to specify.
4. Alter the quantity of or remove the products present in their cart currently.
5. Place their final order. The contents of the entire cart will be ordered on the behalf of the user, but they have been given multiple opportunities to change the contents before that. A warning is displayed before the order is placed as well. The coupon available to them which offers the maximum discount will automatically be applied on the order and a receipt will be generated.
6. View all the coupons available to them.

Relations:

The tables or relations that have been used in the database are described below. The attributes of all the tables have been listed and the primary keys in them have been underlined as per convention.

Admin : Details of admins of various branches (Branch and Admin have a one-one relationship)
User_id, Username, Pass_word, Pincode

Available : Available products at a particular branch
Pincode, Product_id, Quantity

Branch : Names of branches of a pincode which will be used as a foreign key in other tables
Area, Pincode

Cart : Contents of carts of all the customers
User_ID, Product_ID, Quantity

Coupon : (Order_id and order_date are set to null until the coupon is availed)
Coupon_ID, Min_Order_Amt, Valid_Until_Date, Discount_Offered, Date_Of_Issue, Order_id, Order_date, user_id

Customer : Customers' details
User_ID, Name, Contact_no, Total_no_of_orders, pincode

Orders : Description of the orders placed
Order_id, Order_date, Total_price, Delivery_date, Pincode, user_id

Product : Description of the products available at all the branches
Product_id, Product_name, Price

Products_In_Order : Description of the products in the orders in queue
Order_id, User_id, Product_id, quantity

Transactions

T1: Admin A changes the quantity of available products at a branch.

T2: Customer C checks whether products in their cart are available at the nearest branch and then proceed with the order by making an entry in the order table, emptying their cart and deleting the corresponding items from the available list.

T1

Read(Available): Read quantity from available

Input: Take input from user

Write(Available): Write new quantity into available

T2

Read(Cart): Read quantity from cart

Read(Available): Reads quantity in available

Check: Compare quantity in both tables

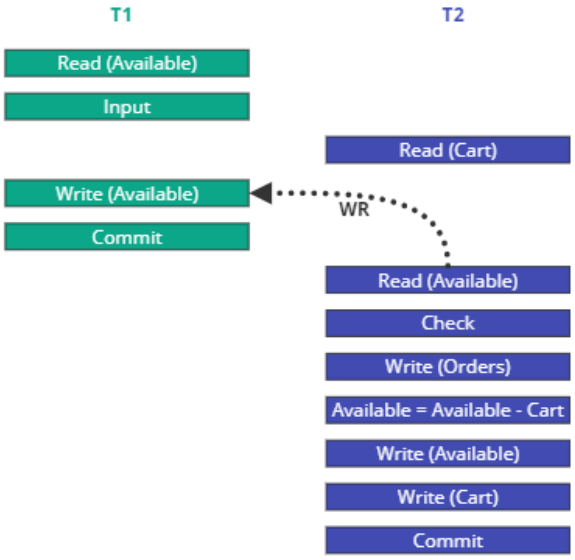
Write(Orders): Create new order entry in orders

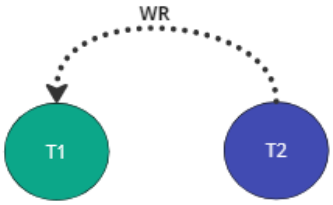
Available := Available - Cart

Write(Available): Updates available table

Write(Cart): Empties cart

Conflict Serializable Schedule

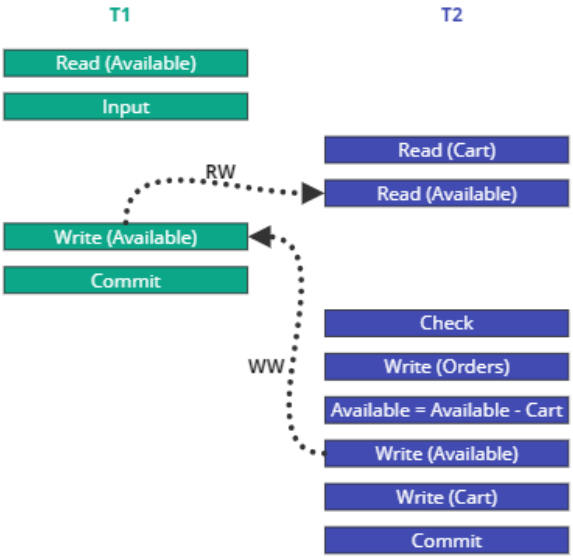


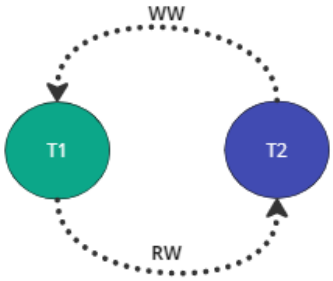


Precedence Graph

We can say that the schedule is conflict serializable because there is no cycle being formed in the precedence graph.

Non-Conflict Serializable Schedule





Precedence Graph

We can say that the schedule is non-conflict serializable because there is no cycle being formed in the precedence graph. We can reduce the non-conflicting schedule.