Introduction to Database Design, Implementation and Administration

Final Project – Point of Sale System

Specifications:

A local supermarket has decided to keep a record of customer purchases for each of its steady Customers. The system should allow the supermarket to track its inventory (inventory is the list of Items that it currently buys from Vendors and sells to Customers). The supermarket managers would like to use to use the system to help ensure that they don’t run out of inventory and can always meet customer demand. The data will eventually be analyzed to identify Customer purchase trends and help the managers compete with the other supermarkets in the area.

Part A:

Design a set of related tables ( a database) that will effectively store the data that is described below, providing easy retrieval and helping to ensure data integrity and validity.

Identify each table’s attributes, PK and FKs (if any)

Specify which fields are mandatory or optional.

Using MySQL Workbench, create an ERD to present your design.

* Each steady Customer is asked to provide the following information: first and last name, address, phone number. A Customer can also choose to store his EBT card ID in his record and the information about one Credit Card. The system will assign each Customer a unique ID. All this data will be stored in the database
* An EBT card is issued by the government and can be charged to cover expenses incurred for food item purchases only. Each month the government adds a certain amount of money to the current balance of the EBT card. The balance is reduced each time a Customer uses the EBT card to pay for a purchase of a food item.
* Each Employee is asked to provide the following information: first and last name, address, phone number, social security number, birthdate. The system will assign each Cashier a unique id and will record the date the Cashier has been hired. The supermarket currently employs Cashiers, Managers and stock clerks (the employees who stock the Item on the shelves of the supermarket). A Cashier will enter his/her id when processing a sale to a Customer.
* Each Vendor (companies that supply the supermarket with products) is identified by its name, phone number. The system will assign a unique vendor id to each Vendor.
* For each Item that the supermarket carries, the system will store the Item Universal Product Code (UPC) , the name of the Item, unit price, product category (ALUMINUM, BAKERY, CANDY, CHICKEN, FISH, FRUIT, GROCERY, MAGAZINES, MEAT, PAPERGOODS), whether it is TAXABLE (Y/N), whether it is FOOD (Y/N), quantity in inventory, and reorder level.
* Sometimes the supermarket applies a discount to the price of certain Items for a given period of time. For each DiscountedItem the supermarket records the upc of the Item, the start date and end date of the sale, the quantity limit if one applies, and the discounted price. The supermarket may require a minimum purchase of non-sale Items to be eligible for the discounted price.
* Each time the supermarket wishes to replenish its inventory of an Item, it compiles a PurchaseOrder from the Vendor that supplies that/those Item(s). The PurchaseOrder will store the date, vendorid and totalDue. A PurchaseOrder can consist of one or more Lineitems. Each line item contains the product code, quantity ordered, unit cost (negotiated with and provided by the vendor), subtotal.
* Each time the supermarket receives a ReceiptofGoods from a Vendor, it records the quantity received and updates the quantity in inventory for each Item received. Each ReceiptofGoods is associated with one or more PurchaseOrders. The Item quantity ordered may be more or less than the quantity received; therefore quantity received must be recorded for each PurchaseOrder Lineitem.
* Each time a Cashier processes a Sale to a Customer, a SalesOrder is stored in the database. The SalesOrder includes a unique sales order ID, date of sale, customerID if available, cashierID, total sale. For each SalesOrder there can be one or more SalesOrderDetail records. Each SalesOrderDetail includes the UPC of the Item sold, the quantity sold, the unit price, whether it was on sale at the time of Sale.
* The Customer can pay for each SalesOrder with a CreditCard, EBT card, cash, DebitCard, and/or Check or a combination thereof. The Payment is recorded and the PaymentDetails are recorded as well.
* The Customer is presented with a Receipt (see sample on last page of this document) that reflects the details of the Sale. Each Receipt is associated with a specific SalesOrder.
* The system records the current sales tax that is applied to each taxable item. When the government modifies the sales tax rate, the updated sales tax is inserted into the system and the date from which it applies is recorded with the sales tax. When a taxable Item is sold, the latest tax rate is applied. When a Customer returns an Item the system can verify how much sales tax was applied at the time of the Sale by verifying the date of the Sale that appears on the Customer Receipt.
* A Customer can Return an Item. The Manager can decide whether the Item can be returned in its present condition. If the Manager decides to allow the Customer to return the Item, the Customer is refunded the amount he paid for the Item at the time of the Sale. This can be determined by looking up the SalesOrderDetail associated with the SalesOrder that is associated with the Receipt the Customer presents when he returns the Item
* The supermarket will record each Payment it makes to a Vendor to pay for goods received from the Vendor. A Payment should be associated with a PurchaseOrder.

Part B:

Using SQL Server, implement the design you have outlined in Part A.

Each table should include necessary attributes, a PK and FK(s) if applicable

Include calculated fields when appropriate

Include constraints to help ensure data validity. For example, a unit price of an Item can’t be less than or equal to zero.

Include triggers that calculate totals in related tables for example, sum the total of all PurchaseDetails and update totalDue in the related PurchaseOrder. (optional)

Part C:

Write a sql statement to address each of the following requests for information:

1. For each Customer, list customer name, address , phone number
2. For each category of item, list the category and the amount of sales that has been generated
3. For each Item, list the upc, description, and how much income was generated. Income will be the quantity purchased \* unit cost minus (quantity sold \* unit price)
4. For each item, list the item, description, vendor name and the number of times its price has been discounted.
5. For a given date, list which items’ prices have been discounted on that date.
6. Which item has been purchased the most often?
7. Which cashier has rung up the largest total of sales?
8. How many items are supplied by each Vendor?
9. List the upc, description, vendor name of each item for which the quantity in stock is less that reorder level
10. What is the description of the item that is the most expensive item (unit price) in inventory?
11. List each EBT card on file, the name of the Customer, current balance on the card, and when (date) it was last used to pay for a Sale
12. Which customer has generated the most sales?
13. Which category of item has generated the largest $ amount of sales?
14. For each vendor list the vendor information and the description of each Item that the vendor provides.
15. List the SalesOrders that were sold without identifying a customer
16. From which Vendor(s) have no purchase orders been generated during the current month.
17. To which Customers have no sales orders been generated during the last 30 days.
18. List the names of Customers who have purchased items of both categories of MEAT and FISH.
19. List the names of each Vendor who supplies ***all the same categories*** of items as Vendor ‘X’. (not an easy query)
20. For each item purchased, list how many times this item has been returned by any customer.
21. List the total sales of food items, how much was paid for by EBT and how much by other means.

A close up of a piece of paper

Description automatically generated