CSE 4126

DISTRIBUTED DATABASE MANAGEMENT SYSTEM

PROJECT REPORT

GO Cart

SUBMITTED BY

TONMOY SINHA | 15-01-04-129

GROUP MEMBERS

AREFEEN SULTAN | 15-01-04-111
SAYED HOSSAIN KHAN | 15-01-04-133

Project Summary:

A super shop is a place where people can buy all their daily necessary products. Nowadays in our country, it has become a new trend to start business. People usually go to super shop and buy their daily need products. But if there is a guidance, then it becomes more useful for them. Our goal is to make shopping more serviceable and reliable. From this thinking we thought of developing a management system using distributed database management system. Here are some of the features of our management system :

1. Customer can buy product and local store can sell products
2. Customer can be a member
3. There will be searching option for customers for information
4. Warehouse can distribute product to local store
5. Warehouse has product inventory
6. Customer can search in different stores for availability of product
7. Customer can see top selling products too
8. There will be monthly, weekly and daily sales report

Platforms:

- Programming Language : PL/SQL
- IDE : Oracle 10g

Entity Relationship Diagram (ERD)

An entity relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities.

Elements of ERD:

- Entities
- Relationships
- Attributes

Entity: An entity is an object or concept about which you want to store information.

Relationship: Relationship shows how entities share information in the database.

Attributes: An attribute refers to a database component.

Entity Sets:

- Customer
- Branch
- Membership
- Product
- Transaction
- Warehouse
- Category

Entity and Table Set Name with Attributes and data types

Attributes of Branch:

- Branch_ID(primary key) - int
- Location - varchar(11)
- Phone - int

Attributes of Category:

- C_ID(primary key) – int
- C_Name – varchar(20)

Attributes of Customer:

- Cust_ID(primary key) - int
- M_ID(foreign key) - int
- C_Name - varchar(50)

- Email - varchar(50)

Attributes of Membership:

- M_ID(primary key) - int
- P_Range_From - number
- P_Range_To - number
- Discount_Rate – number
- Type - varchar(50)

Attributes of Product:

- P_ID(primary key) - int
- C_ID(foreign key)- int
- Selling_Price – number
- P_Name – varchar(50)

Attributes of Transaction:

- T_ID(primary key) - int
- Cust_ID(foreign key) – int
- Branch_ID(primary key) – int
- Total_Price – number
- Date – date

Attributes of Warehouse:

- S_ID(primary key) - int
- S_Date - date
- P_ID(foreign key) – int
- Buying_Price - number
- P_Quantity – int

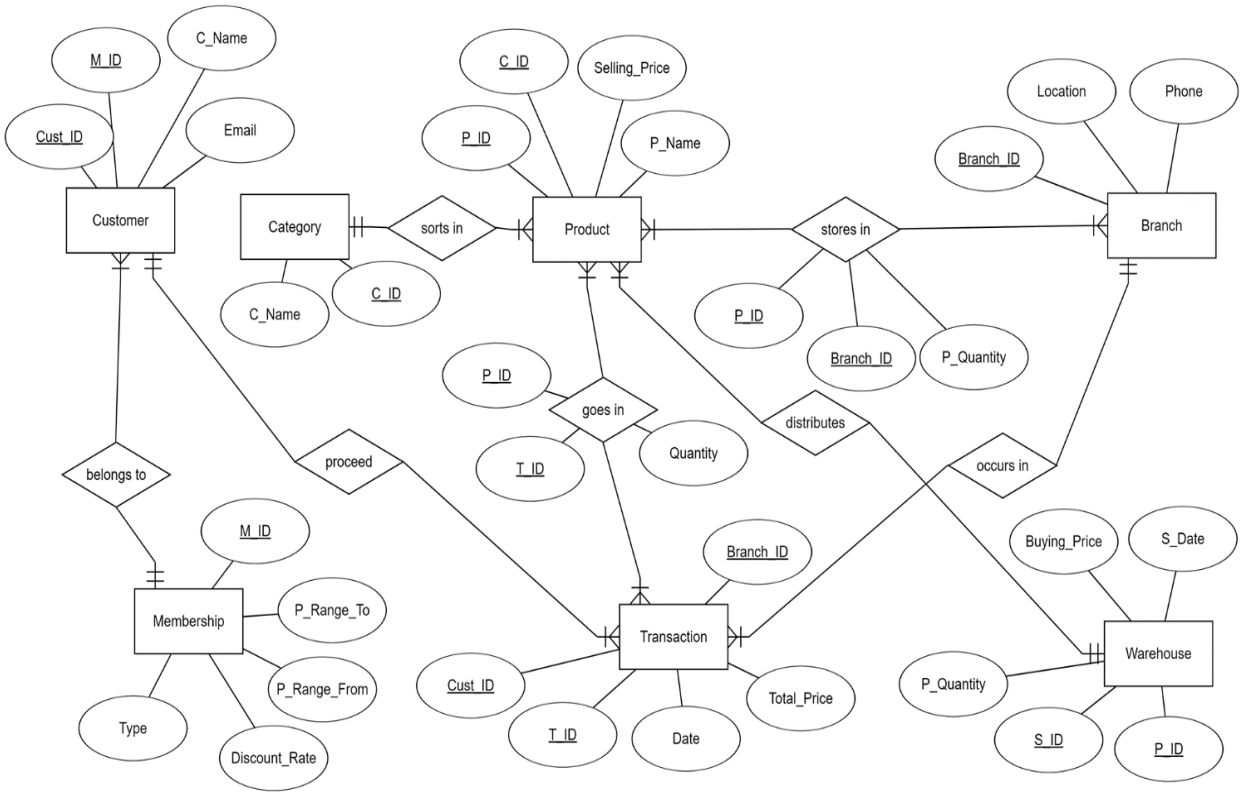
Attributes of goes in:

- P_ID(foreign key) - int
- T_ID(foreign key) – int
- Quantity – int

Attributes of stores in:

- P_ID(foreign key) - int
- Branch_ID(foreign key) – int
- P_Quantity – int

Entity Relationship Diagram (ERD):



Relational Model with Fragmentation Schema:

Global Schema:

- Branch(branch_ID, Location, Phone)
- Category(C_ID, C_Name)
- Membership(M_ID, P_Range_From, P_Range_To, Discount_Rate, Type)
- Product(P_ID, Selling_Price, P_Name, C_ID)
- Customer(C_ID, C_Name, Email, M_ID)
- Transaction(T_ID, Cust_ID, Branch_ID, Total_Price, T_Date)
- goes_in(P_ID, T_ID, Quantity)
- occurs_in(Branch_ID, T_ID)
- proceed(T_ID,C_ID)
- stores_in(P_ID, P_Quantity, Branch_ID)
- warehouse(S_ID, S_Date, P_ID, P_Quantity, Buying_Price)
- Sales(Serial_no, Month, Sales)

Fragmentation Schema with allocated site:

- Branch1 = $PJ_{branch_ID,Location,Phone}SL_{Location='Dhanmondi'}$Branch
- Branch2 = $PJ_{branch_ID,Location,Phone}SL_{Location='Mohammadpur'}$Branch
- Stores_in1 = $PJ_{P_ID,\ P_Quantity,\ Branch_ID}SL_{Branch_ID\ =\ 1}$stores_in
- Stores_in2 =$PJ_{P_ID,\ P_Quantity,\ Branch_ID}SL_{Branch_ID\ =\ 2}$stores_in
- Transaction1 =$PJ_{T\_ID,Cust\_ID,Branch\_ID,Total\_Price,T\_Date}SL_{Branch\_ID\ =\ 1}$ Transaction
- Transaction2 = $PJ_{T\_ID,Cust\_ID,Branch\_ID,Total\_Price,T\_Date}SL_{Branch\_ID\ =\ 2}$ Transaction
- Branch1 @ site1, Stores_in1 @ site1, Transaction1 @ site1
- Branch2 @ site2,Stores_in2 @ site2, Transaction2 @ site2

Database Profile:

For relation Branch (branch_ID, Location, Phone) with fragments $Branch_1$, $Branch_2$ the database profile contains the following information:

$Branch_1$

$card(Branch_1) = 1$

$site(Branch_1) = 1$

	Branch_ID	Location	Phone
Size	16 bits	11 bits	16 bits
Val	1	1	1

$\underline{Branch_2}$

card($Branch_2$) = 1

site($Branch_2$) = 2

	Branch_ID	Location	Phone
Size	16 bits	11 bits	16 bits
Val	1	1	1

For relation Stores_in(P_ID, P_Quantity, Branch_ID) with fragments $Stores\_in_1$, $Stores\_in_2$ the database profile contains the following information:

$\underline{Stores\_in_1}$

card($Stores\_in_1$): 1

site($Stores\_in_1$): 1

	P_ID	P_Quantity	Branch_ID
Size	16 bits	16 bits	16 bits
Val	1	1	1

$\underline{Stores\_in_2}$

card($Stores\_in_2$): 2

site($Stores\_in_2$): 2

	P_ID	P_Quantity	Branch_ID
Size	16 bits	16 bits	16 bits
Val	2	2	1

Transaction$_1$

card(Transaction$_1$): 2

site(Transaction$_1$): 1

	T_ID	Cust_ID	Branch_ID	Total_Price	T_Date
size	16 bits	16 bits	16 bits	21 bytes	7 bytes
val	2	2	1	2	2

Transaction$_2$

card(Transaction$_2$): 2

site(Transaction$_2$): 2

	T_ID	Cust_ID	Branch_ID	Total_Price	T_Date
size	16 bits	16 bits	16 bits	21 bytes	7 bytes
val	2	2	1	2	2

Functionsor Procedures:

1. **Top Selling Product:**We can see the top selling products of a week,a month or a year.
2. **Sales Report:**Sales report will be produced based on buy and sales.
3. **Warehouse Distribution:**Distribution data of the products on each branch will be kept in.
4. **Product availability:**Product availability of nearby branches will be informed to the customer.
5. **Membership:**Customer will get discount based on membership rank.

Effect of Update:

Update Branch_ID = 2 where product ID = 10001:

- 1st step:

Stores_in1

Product ID	Product Quantity	Branch ID
10001	3	1

Product ID	Product Quantity	Branch ID
10002	1	2
10003	2	2

Stores_in2

- 2nd Step:

Deletes 10001 product id from Stores_in1

Stores_in1

Product ID	Product Quantity	Branch ID
10001	3	1

Stores_in2

Product ID	Product Quantity	Branch ID
10002	1	2
10003	2	2

- 3rd Step:

Insert into Stores_in2 @ site 2:

Stores_in1

Product ID	Product Quantity	Branch ID

Product ID	Product Quantity	Branch ID
10002	1	2
10003	2	2
10001	3	2

Stores_in2

Operator Tree on Join:

Query: SL_{C_Name}(Category $JN_{CID=CID}$ Product)

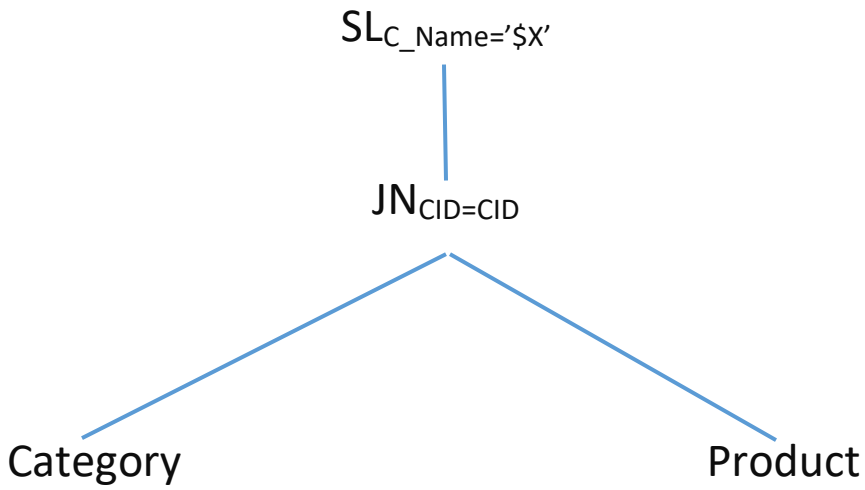


Fig: Operator Tree

Qualified Relation:

$SL_{Cust\_ID=1001}$[Transaction1: Branch_ID = 1]

Applying Rule 1,

⇨ [$SL_{Cust_ID=1001}$Transaction1: Branch_ID = 1 and Cust_ID=1001]

Transaction:

T_ID	Cust_ID	Brach_ID	Total_Price	T_Date
1	1001	1	500	09-JAN-2018
2	1002	2	36000	02-JAN-2018
3	1003	2	28500	03-JAN-2018
4	1004	2	72000	04-JAN-2018
5	1005	1	16000	02-JAN-2018
6	1006	2	19000	05-JAN-2018
7	1007	2	1050	08-JAN-2018
8	1008	1	28000	01-JAN-2018
9	1009	2	56000	14-JAN-2018

Transaction1:

T_ID	Cust_ID	Brach_ID	Total_Price	T_Date
1	1001	1	500	09-JAN-2018

Semi-join Program:

In procedure 5 our query was:

PJ_{Type} $SL_{C\_ID='\$x'}$ (Membership $JN_{M\_ID = M\_ID}$ Customer).

So for the equivalence, semi-join program would be:

⇨ PJ_{Type} $SL_{C\_ID='\$x'}$((Membership $SJ_{M_ID = M_ID}$$PJ_{M_ID}$Customer) $JN_{M_ID=M_ID}$Customer).

We can ensure that the semi-join program outputs the same result as join program.

Contribution :

As it was a group project, so some of the works were done in group. I couldn't contribute much in the project as I was sick. But I tried to help my teammates. Here are some of my contributions in the project :

- Create the dataset
- Create some of the fragments
- Help to make the functions
- Contribute later to make the database profile
- Contribute to implement machine learning algorithm