

Ahsanullah University of Science & Technology
Department of Computer Science & Engineering



Course No: CSE4126
Course Title: Distributed Database Lab
Project Title: Al-Bintum Super Shop

Submitted to:
Mohammd Imrul Jubair
Assistant Professor, AUST

Submitted by:
Md. Saiful Islam || Id: 15.01.04.027

Other Members:
Al Farabi Akash || Id: 15.01.04.010
Oishee Binte Haque || Id: 15.01.04.05

Project Summary:

The main purpose of our project is to maintain a perfect, accurate and mostly dynamic system for managing a super shop. Our super shop management system will process the each and every part related to a super shop like sales, staffs, branch information and the updates based on sales. From knowing all this things at a quickest possible time the managing process will get more boost up for the future how it should go further.

Objectives:

1. To maintain the all branches info dynamically
2. To maintain the staffs info against each branch
3. To know about the product is available or not.
4. Maintaining the payment history of customers
5. Gathering info of products (after sell)

Entity Relationship Diagram (ERD):

The whole system at a glance:

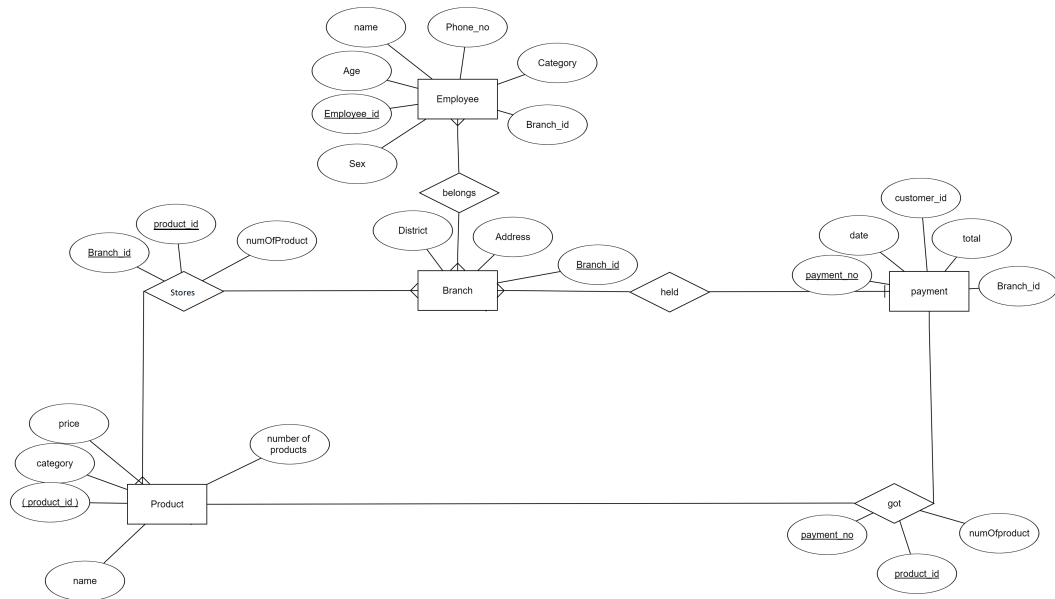


Figure-1: ERD of Super Shop Management System

Global Relations:

1. **EMPLOYEE** (employeeId, ename, sex, age, phoneNo, catagory, branchId)
2. **BRANCH** (branchId, address, district)
3. **PRODUCT** (productid, productname, price, category, numberofproducts)
4. **STORES** (productid, branchId, numberofproducts)
5. **GOT** (paymentno, productid, noOfProducts)
6. **PAYMENT** (paymentno, branchid, total, paymentdate)

Fragmentation Schema:

We have used following fragments:

Employee1 = $SL_{branchid=1}(Employee)$ Employee2 = $SL_{branchid!=1}(Employee)$
Stores1 = $SL_{branchid=1}P_{j_{productid, branched, numberOfProducts}}(Stores)$ Stores2 = $SL_{branchid!=1}P_{j_{productid, branched, numberOfProducts}}(Stores)$
Branch1 = $SL_{branchid=1}(Branch)$ Branch2 = $SL_{branchid!=1}(Branch)$
Payment1 = $SL_{branchid=1}(Payment)$ Payment2 = $SL_{branchid!=1}(Payment)$

Allocation Schema:

Site1(Host): Stores1,Product,Branch1,Employee1, Payment1,Got	Site2: Stores2,Branch2,Employee2, Payment2
---	---

Fragmentation Tree:

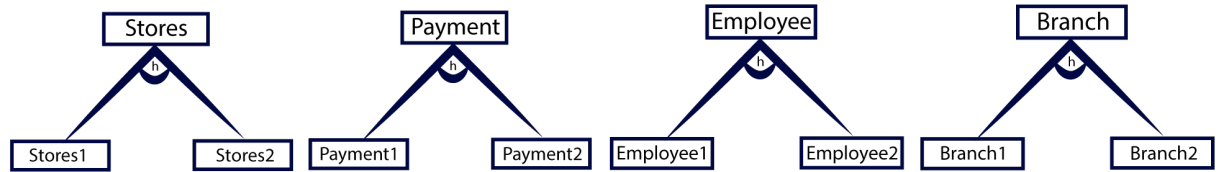


Figure-2: Fragmentation Trees

Allocation Schema (Visual Image):

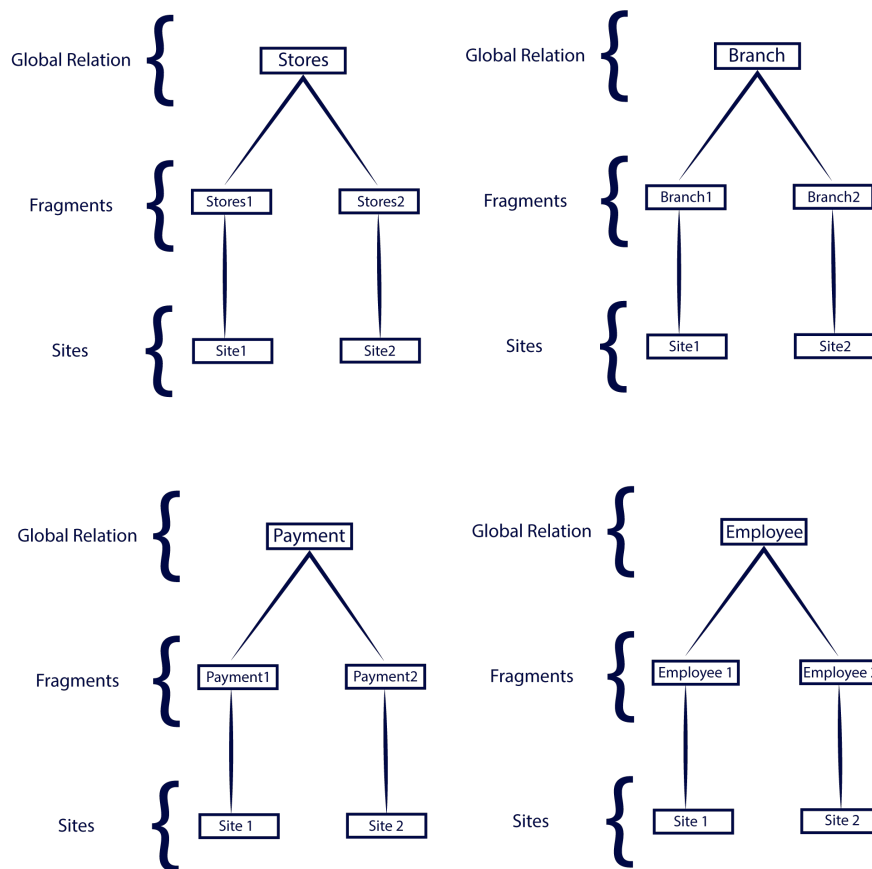


Figure-3: Allocation Schema (visualization)

Database Connection:

In terms of connecting sites we used 2 laptops. One was for site1 and another is for site2. Firstly, (site1) we inserted data tables and then turned off the windows defender. The static IP here is 192.168.43.101 .Then we pinged from site2 (static IP 192.168.43.102). Both of them were in the same network. We got the ping status then. We did the following changes in site2:

Added the following line in listener.ora

```
( SID_DESC =  
(SID_NAME = XE)  
(ORACLE_HOME = C:\oracle\app\oracle\product\10.2.0\server)  
)  
  
( ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.43.102)(PORT  
= 1521))
```

After adding lines in cmd we performed lsnrctl stop and lsnrctl start and they both came with successful message.

Finally for Database link at host side:

```
drop database link site_link;  
create database link site_link  
connect to system identified by "somum123"  
using '(DESCRIPTION =  
(ADDRESS_LIST =  
(ADDRESS = (PROTOCOL = TCP)  
(HOST = 192.168.43.102)  
(PORT = 1521))  
)  
(CONNECT_DATA =(SID = XE))  
)',  
;
```

Function, Procedure & Trigger:

Functions:

1. Find out bill for selected products.
2. Creating bill with the option of updating, inserting, and searching.
3. Find out the category of employee.
4. Find out employee against branch.
5. Update employee (in terms of transfer) according to branchid.
6. Update quantity of products by Id and branch.

Procedures:

1. Update price on given range and given percentage.
2. Finding the availability of products.
3. Find out sell details with respect to branchid.

Triggers:

1. Product Price After Update Trigger

Shows the old price, updated price and difference between the price

1. Store Product After Update Trigger:

If any store updates its number of products, this trigger updates the value in its fragmented tables and also shows a message which site got the update

My Contribution:

1. First of all I did the whole part of database connection along with link which is already defined in previous section (Database Connection).
2. Secondly, the part of 2 functions related to finding out the employee and updating the employee against branch.

Scenario for updating employee branch from branch 1 to 2:

Parameter: employeeId number, branchId number

Return: number

Description: takes employeeId,branchId and updates employee table according to branchId.

Exception: NONE

Example Update Process:

Update -> Branch Id = 2 where EmployeeId = 10

- Store corresponding data of employeeid.
- Insert in Employee2 with updated value
- Delete data from employee1

Employee1:

Employee Id	Branch Id	Ename	Sex	Phone	Address
10	1	abcd	F	01XX	DHAKA



Employee2:

Employee Id	Branch Id	Ename	Sex	Phone	Address
10	2	abcd	F	01XX	DHAKA

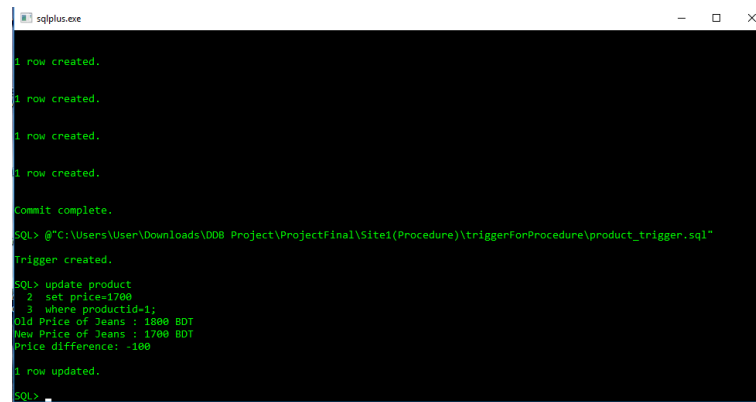


Employee1:

Employee Id	Branch Id	Ename	Sex	Phone	Address
10	1	abcd	F	01XX	DHAKA

3. Lastly, one of trigger in terms of updating product and showing afterwards message.

Result of Updating Process:



```
sqlplus.exe

1 row created.

1 row created.

1 row created.

1 row created.

Commit complete.

SQL> @'C:\Users\User\Downloads\DOB Project\ProjectFinal\Site1(Procedure)\triggerforProcedure\product_trigger.sql'
Trigger created.

SQL> update product
  2 set price=1700
  3 where productid=1;
Old Price of Jeans : 1800 BDT
New Price of Jeans : 1700 BDT
Price difference: -100

1 row updated.

SQL>
```

Figure-4: Update message (with Trigger)

Future Plan:

As we improved a total system for super shop and nowadays it is spreading rapidly so we have the opportunity to develop it for further.

1. Adding more option in database for simplifying (site wise).
2. Creating more usable function/procedure.
3. Using the machine learning process for taking long-term decision.