



# **Ahsanullah University of Science & Technology**

Department of Computer Science and Engineering

Course Title: Database Lab

Course No: CSE 3104

## **Submitted To**

Mr. Mohammad Moinul Hoque

Mr. Mir Tafseer Nayeem

## **Project By**

Raihan Tanvir

ID:16.01.04.140

Susmoy Kar

ID:16.01.04.141

## **Project Name:**

Super Shop Management System

## **Project Objective:**

Super Shop Management System is a simple, yet very effective tool for managing super shop. Through this application shop owner/manager can do the following task:

- **Inventory Management:** Information about products such as price, available units in stock, supplier of products, daily/monthly sale of a product etc.
- **Customer Info:** Details of customer's information can easily be stored, managed & used as necessary.
- **Supplier Info:** Information about suppliers of various product can be stored & managed.
- **Employee Info:** Information about employees of the shop can be stored & managed.

## **Types of Users:**

(a)Normal User

(b)Private User

(c)Admin

## **Feature Grouping According to User:**

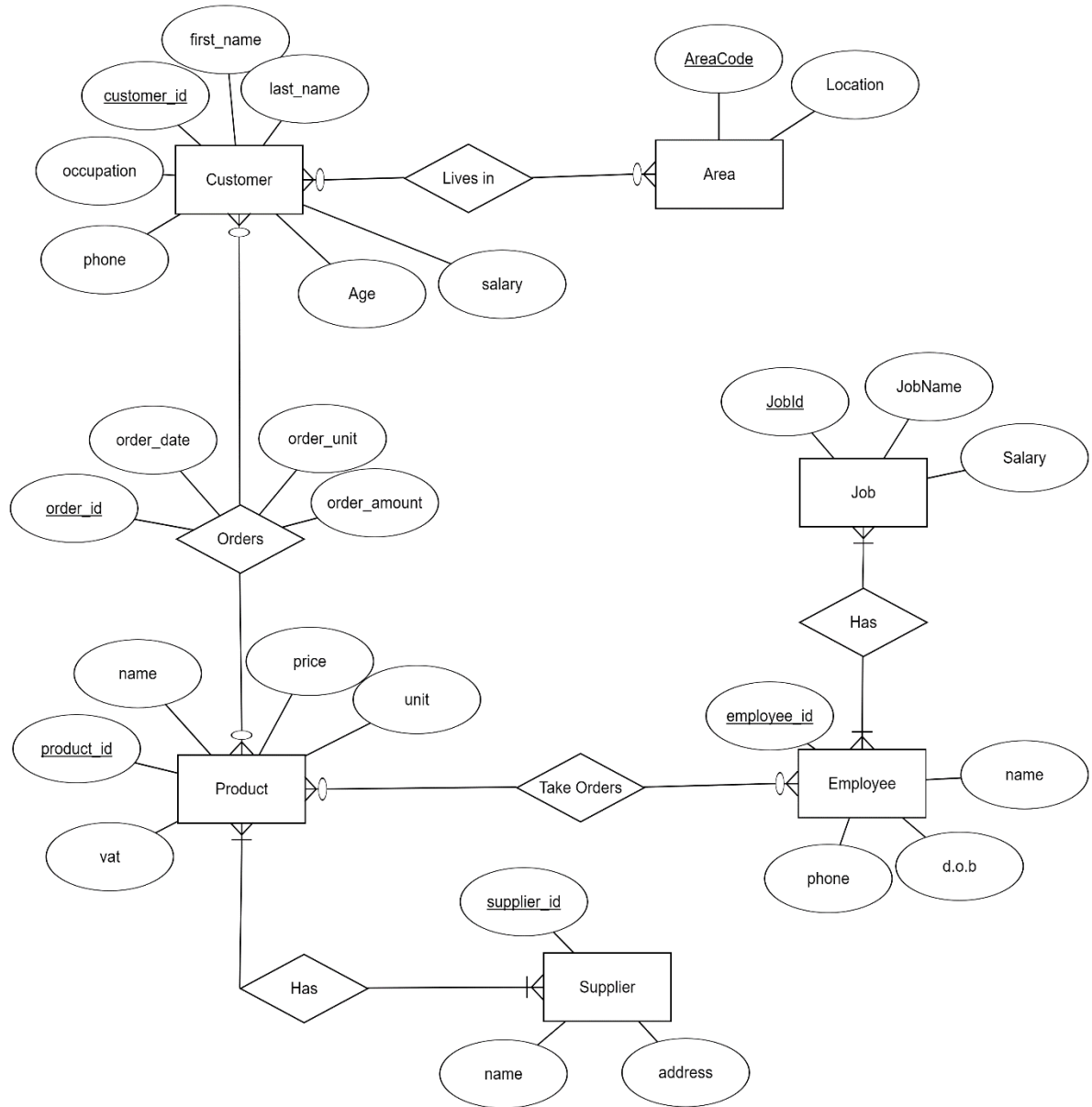
- 1. Normal User:** Normal Users are the customers i.e. public user.
  - a.** They can view the product list available in the inventory with unit price of the products.
- 2. Private User:** Private Users are the employee of the shop.
  - a.** Can view all product related information.
  - b.** Can see all customer related information.

- c. Can find the product with maximum price, 2<sup>nd</sup> maximum price, 3<sup>rd</sup> maximum price...so on.
  - d. Can view customers with a particular name
  - e. Can find customers of a specific mobile operator.
  - f. Can view customers with total amount they spent.
  - g. Can filter customer according to their occupation, salary, age etc.
  - h. Can find the number of available products in the inventory.
- 3. Admin:** Admin is the manager of the shop. He can do the following task:
- a. Can do all the operation of private users.
  - b. Can View all Supplier related information.
  - c. Can see all employee related information.
  - d. Can update product prices.
  - e. Can alter employee's salary.
  - f. Can View total product supplied by each supplier.
  - g. Can see total amount sold by each employee.
  - h. Can hire new employee or terminate old ones.

### **Name of The Entities:**

- 1. Customer: CustomerID
- 2. Area: AreaCode
- 3. Product: ProductID
- 4. Supplier: SupplierID
- 5. Employee: EmployeeID
- 6. Job: JobID

## Entity Relationship Diagram:



## Relational Model:

Create Table Area

```
(  
AreaCode int IDENTITY(1000,1) PRIMARY KEY,  
Address varchar(200),  
)
```

CREATE TABLE CUSTOMER

```
(  
CustomerId int IDENTITY(1,1) PRIMARY KEY,  
FirstName varchar(50) NOT NULL,  
LastName varchar(50) NOT NULL,  
Age int NOT NULL CHECK (Age >= 18),  
Phone varchar(11) NOT NULL,  
Occupation varchar(50),  
Salary decimal(18,2),  
AreaCode int NOT NULL FOREIGN KEY REFERENCES Area(AreaCode),  
)
```

Create Table Job

```
(  
JobId int IDENTITY(1,1) PRIMARY KEY,  
JobName varchar(200),  
Salary money NOT NULL CHECK (Salary >= 1000),  
)
```

CREATE TABLE EMPLOYEE

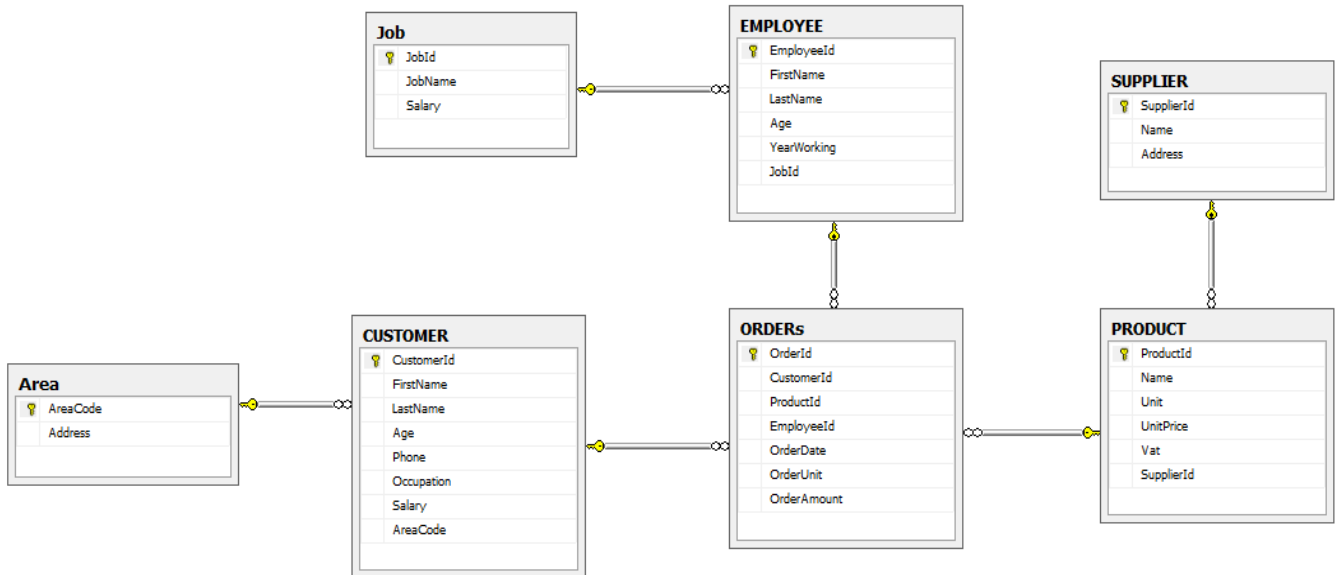
```
(  
EmployeeId int IDENTITY(1,1) PRIMARY KEY,  
FirstName varchar(50) NOT NULL,  
LastName varchar(50) NOT NULL,  
Age int NOT NULL CHECK (Age >= 18),  
YearWorking int NOT NULL,  
JobId int NOT NULL FOREIGN KEY REFERENCES Job(JobId),  
)
```

```
CREATE TABLE SUPPLIER
(
SupplierId int IDENTITY(1,1) PRIMARY KEY,
Name varchar(50) NOT NULL,
Address varchar(50),
)
```

```
CREATE TABLE PRODUCT
(
ProductId int IDENTITY(1,1) PRIMARY KEY,
Name varchar(50) NOT NULL,
Unit int NOT NULL,
UnitPrice int NOT NULL,
Vat int NOT NULL,
SupplierId int NOT NULL FOREIGN KEY REFERENCES Supplier(SupplierId),
)
```

```
CREATE TABLE ORDERs
(
OrderId int IDENTITY(1,1) PRIMARY KEY,
CustomerId int NOT NULL FOREIGN KEY REFERENCES CUSTOMER
(CustomerId),
ProductId int NOT NULL FOREIGN KEY REFERENCES Product (ProductId),
EmployeeId int NOT NULL FOREIGN KEY REFERENCES Employee
(EmployeeId),
OrderDate date NOT NULL,
OrderUnit int NOT NULL,
OrderAmount money NOT NULL,
)
```

## Database Diagram:



## SQL Queries:

```
INSERT INTO Area(Address)
VALUES ('Banani'),('Gulshan'),('Basabo'),('Uttara'),('Khilgaon')
```

```
INSERT INTO
CUSTOMER(FirstName,LastName,Age,Phone,Occupation,Salary,AreaCode)
VALUES ('Rahim','Uddin',35,'01832702340','Doctor',80000,1),
('Karim','Uddin',31,'01732742340','Engineer',120000,2),
('Maruf','Khan',45,'01932502125','Teacher',60000,1),
('Abir','Ahmed',22,'01516781648','Student',15000,3)
```

```
INSERT INTO Job(JobName,Salary)
VALUES
('Salesman',15000),('Counterman',20000),('Cleaner',10000),('StoreKeeper',20000)
```

```
INSERT INTO EMPLOYEE(FirstName,LastName,Age,YearWorking,JobId)
VALUES ('Alim','Uddin',25,3,1),
```

```
('Steve','Smith',21,2,1),
('David','Warner',35,3,2),
('Dane','Smith',31,2,3),
('Jane','Warne',39,4,4)
```

```
INSERT INTO SUPPLIER(Name,Address)
VALUES ('SUMASHTECH','Farmgate'),
('Uniliver','Banani'),
('Momen Int.','Karwan'),
('AKIZ','Kuril')
```

```
INSERT INTO PRODUCT(Name,Unit,UnitPrice,Vat,SupplierId)
VALUES ('Rice',5000,50,0,3),
('Potato',2000,20,0,1),
('Ice-cream',50,250,12,2),
('Perfume',200,500,35,1),
('Meat',400,400,25,4),
('Milk',200,70,5,4)
```

```
INSERT INTO
ORDERs(CustomerId,ProductId,EmployeeId,OrderDate,OrderUnit,OrderAmount)
VALUES (1,1,1,'02-15-2018',30,1500),
(1,2,1,'02-15-2018',10,200),
(2,3,2,'08-30-2018',2,500),
(3,4,1,'12-11-2018',3,1500),
(4,5,3,'07-11-2018',5,1200),
(1,3,2,'02-15-2018',10,200),
(2,1,3,'09-21-2018',10,500)
```

```
-----
--Function of Normal User(Customer)
-----
```

```
--1
SELECT Name,UnitPrice FROM PRODUCT WHERE Unit>0
```



-----  
--Function of Private User(Employee)  
-----

--0

INSERT INTO

CUSTOMER(FirstName,LastName,Age,Phone,Occupation,Salary,AreaCode)

VALUES ('Rahim','Uddin',35,'01832702340','Doctor',80000,1),

('Karim','Uddin',31,'01732742340','Engineer',120000,2),

('Maruf','Khan',45,'01932502125','Teacher',60000,1),

('Abir','Ahmed',22,'01516781648','Student',15000,3)

--Registering New Customers

--1

INSERT INTO PRODUCT(Name,Unit,UnitPrice,Vat,SupplierId)

VALUES ('Rice',5000,50,0,3),

('Potato',2000,20,0,1),

('Ice-cream',50,250,12,2),

('Perfume',200,500,35,1),

('Meat',400,400,25,4),

('Milk',200,70,5,4)

--Adding Product Into Inventory

--2

INSERT INTO

ORDERs(CustomerId,ProductId,EmployeeId,OrderDate,OrderUnit,OrderAmount)

VALUES (1,1,1,'02-15-2018',30,1500),

(1,2,1,'02-15-2018',10,200),

(2,3,2,'08-30-2018',2,500),

(3,4,1,'12-11-2018',3,1500),

(4,5,3,'07-11-2018',5,1200),

(1,3,2,'02-15-2018',10,200),

(2,1,3,'09-21-2018',10,500)

--Taking Orders from customer

--3

SELECT \* FROM CUSTOMER

--4

```
SELECT * FROM CUSTOMER WHERE AreaCode=(SELECT AreaCode From  
Area WHERE Address='BANANI')
```

--Display info of customers from a particular address

--5

```
SELECT FirstName+' '+LastName AS CustomerName,Occupation,Phone FROM  
CUSTOMER
```

```
WHERE Salary BETWEEN 50000 AND 100000
```

--Display info of customers based on range of salary earned by customers

--6

```
SELECT * FROM CUSTOMER WHERE Age IN (25,35,45)    --Display info of  
customers who are in given age option
```

--7

```
SELECT * FROM CUSTOMER WHERE Occupation='Teacher'
```

--Display info of customers who of a particular occupation like teacher,doctor,buisenessmen etc.

--8

```
SELECT * FROM PRODUCT
```

--Display all product

--9

```
SELECT TOP 1 * FROM PRODUCT ORDER BY UnitPrice DESC
```

--Display product with maximum price

--10

```
SELECT TOP 1 * FROM PRODUCT
```

```
WHERE UnitPrice<(SELECT MAX(UnitPrice) FROM PRODUCT)
```

--Display product with 2nd maximum price

--11

```
SELECT TOP 1 * FROM PRODUCT
```

```
WHERE UnitPrice<(SELECT TOP 1 * FROM PRODUCT
```

```
WHERE UnitPrice<(SELECT MAX(UnitPrice) FROM PRODUCT))
```

--Display product with 3rd maximum price

--12

```
SELECT COUNT(ProductId) FROM PRODUCT
```

--Display Number of products

--13

```
SELECT FirstName+' '+LastName AS CustomerName,Occupation,OrderAmount  
FROM CUSTOMER C,ORDERs O
```

```
WHERE C.CustomerId=O.CustomerId AND
```

```
OrderAmount=(SELECT MAX(OrderAmount) FROM ORDERs)
```

--Display Name,Occupation of customers who ordered product with maximum amount of bill

--14

```
SELECT * FROM CUSTOMER
```

```
WHERE CustomerId IN (SELECT CustomerId FROM ORDERs
```

```
WHERE ProductId=(SELECT ProductId FROM PRODUCT
```

```
WHERE Name='RICE'))
```

--Display customers who ordered a particular product like  
rice,flower,jam,electronics etc

--15

```
SELECT FirstName+' '+LastName AS CustomerName,Age,Phone FROM  
CUSTOMER WHERE FirstName LIKE 'Ka%' OR LastName LIKE '__d%'
```

--16

```
SELECT * FROM CUSTOMER WHERE Phone LIKE '017%'
```

--Display customer who uses GP operator

-----  
--Function of Admin(Manager)  
-----

--1

```
INSERT INTO EMPLOYEE(FirstName,LastName,Age,YearWorking,JobId)  
VALUES ('Adam','Jhonson',25,3,1),
```

```
('Steve','Donald',21,2,1),
```

```
('David','Hussey',35,3,2)
```

--Hiring New Employee

--2

```
INSERT INTO SUPPLIER(Name,Address)
VALUES ('KRY','Farmgate'),
('Oceania','Banani'),
('GnG Int.','Karwan'),
('AFBL','Kuril')
```

--Adding New Suppliers

--3

```
SELECT * FROM EMPLOYEE
--Display all employees info
```

--4

```
SELECT EmployeeId,FirstName+' '+LastName,Age FROM EMPLOYEE
WHERE Age>=30
```

--Display employee name,age whose age is greater or equal 30yr

--5

```
UPDATE Job SET Salary=Salary+Salary*.1 WHERE JobName='Counterman'
--Increment Employee Salary by 10% who are working as counterman
```

--6

```
SELECT EmployeeId,FirstName+' '+LastName,JobName FROM
EMPLOYEE,Job WHERE JobName='Salesman'
```

--Display employee name,designation who are working as salesman

--7

```
SELECT J.JobId,J.JobName,COUNT(EmployeeId) FROM Job J,EMPLOYEE
GROUP BY J.JobId,JobName
```

--Display number of employees departmentwise

--8

```
UPDATE PRODUCT SET UnitPrice=UnitPrice-.05*UnitPrice
WHERE SupplierId=(SELECT SupplierId FROM SUPPLIER WHERE
Name='SUMASHTECH')
```

--Reduce Product price by 5% from a particular Supplier

--9

```
SELECT * FROM PRODUCT WHERE Vat<=100
```

--10

SELECT \* FROM SUPPLIER

--Display all supplier related info

--11

SELECT S.SupplierId,S.Name,COUNT(ProductId) FROM PRODUCT  
P,SUPPLIER S

WHERE S.SupplierId=P.SupplierId GROUP BY S.SupplierId,S.Name

--display number of products supplied by each supplier

--12

SELECT AVG(Salary) FROM CUSTOMER

--12

SELECT JobName,AVG(Salary) FROM Job GROUP By JobName

--avg salary of employee by department

--13

DELETE FROM EMPLOYEE WHERE EmployeeId=3

--delete employee whose id is 3

--14

SELECT E.EmployeeId,FirstName+' '+LastName as

EmployeeName,SUM(OrderAmount)

FROM EMPLOYEE E,ORDERs O

WHERE E.EmployeeId=O.EmployeeId

Group By E.EmployeeId,E.FirstName,E.LastName

--Display total amount of products sold by each employee

**Limitations:**

1. Payment method is not available in the system.
2. In sufficient operation for normal user i.e. customers.
3. Suppliers transactions are not stored.

**Future Scope:**

4. Payment via online financial system like BKash, Roker, UCash and payment gateway like VISA, Master Card, Nexus etc.
5. Keeping Track of payment to each supplier.
6. Converting the existing Shop Management System into a Chain Shop Management System to maintain chains of super shop of same owner.

**Conclusion:**

Theoretical Knowledge gained in sessional is applied in this project. It's an amazing opportunity to demonstrate the knowledge and skill we possess in the field of database. Though it may have some limitations but maximum effort was put behind this project.