

## Lab 1 - Hierarchical Database

Ashwin Shanmugam (UCID: 30300738) & Jasneet Singh (UCID: 30044332)

### Task 1 – Identify Entities and Attributes

SUPERSTORE - Top Level Entity (SuperstoreID, Name, Location)

BRANCH - (BranchId, BranchName, ManagerName)

DEPARTMENT (DepartmentID, DepartmentName, DepartmentManager)

PRODUCT (ProductId, ProductName, ProductDescription, ProductPrice,  
ProductSupplier, UnitsAvailable, ProductExpiryDate)

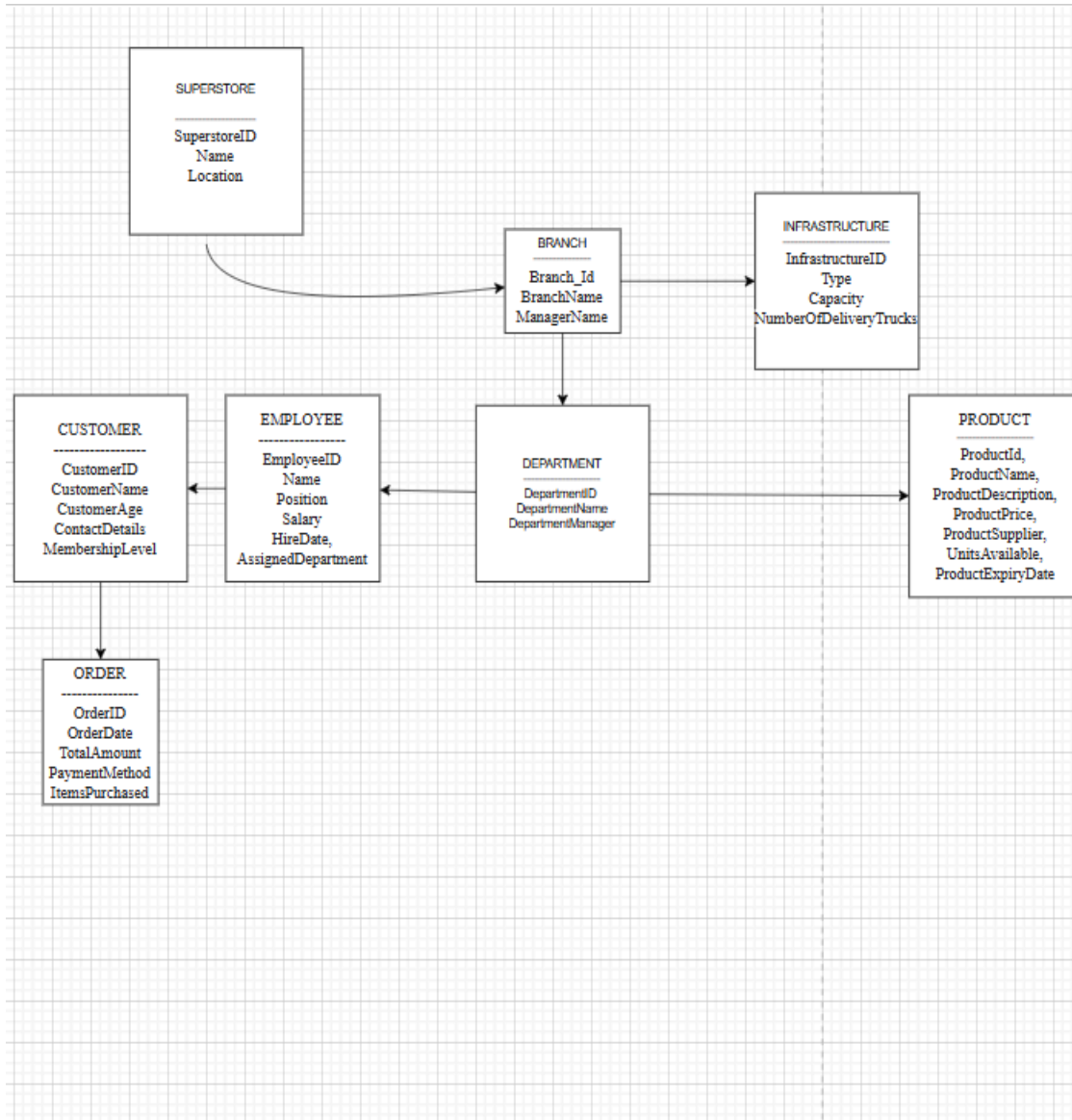
EMPLOYEE (EmployeeID, Name, Position, Salary, HireDate, AssignedDepartment)

CUSTOMER (CustomerID, CustomerName, CustomerAge, ContactDetails,  
MembershipLevel)

ORDER (OrderID, OrderDate, TotalAmount, PaymentMethod, ItemsPurchased)

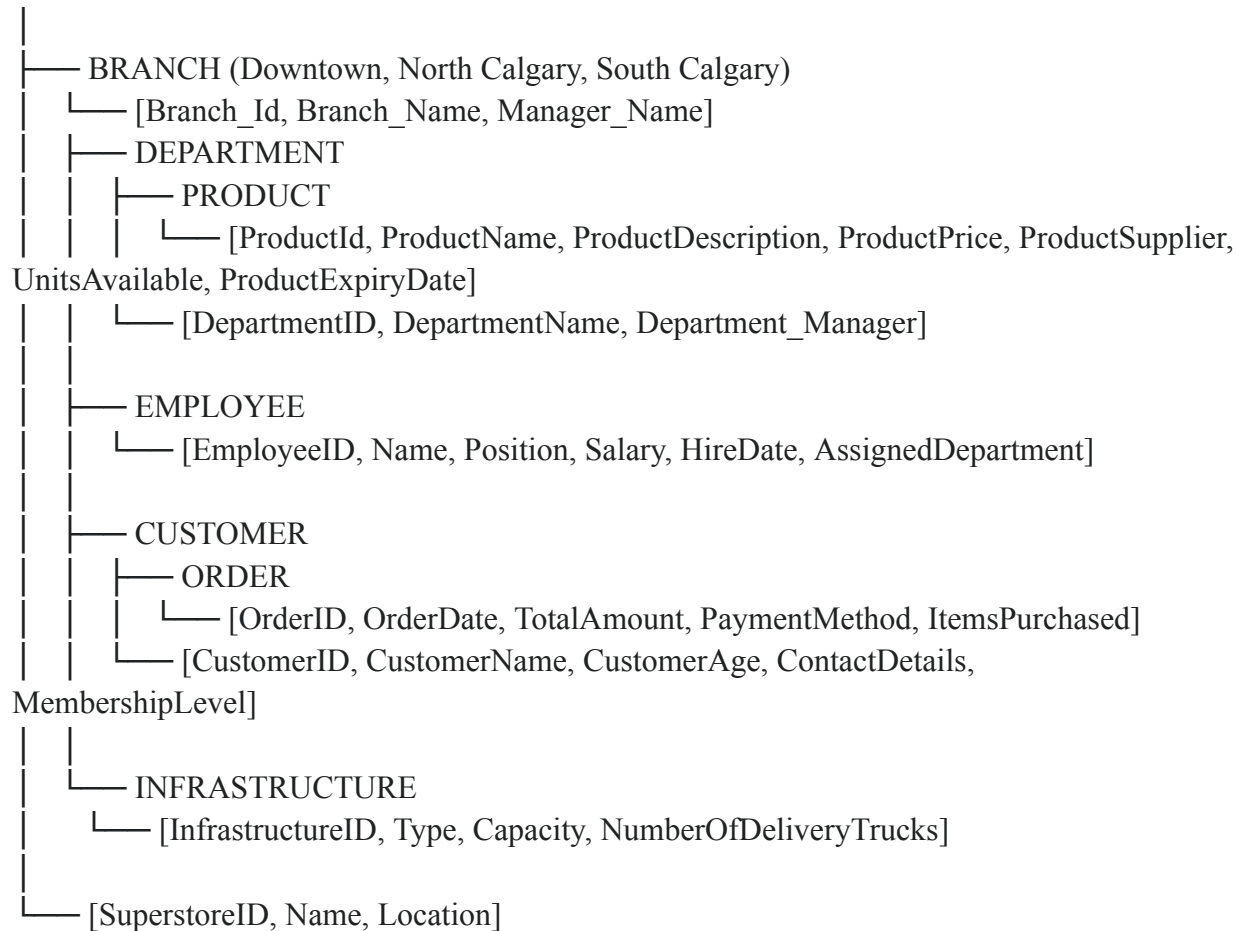
INFRASTRUCTURE (InfrastructureID, Type, Capacity, NumberOfDeliveryTrucks)

## Task 2 – Build the Block Diagram



### Task 3 – Write the Tree Structure

#### SUPERSTORE



### Task 4 – Define Navigational Queries

1. SUPERSTORE → BRANCH → DEPARTMENT → PRODUCT
2. SUPERSTORE → BRANCH → CUSTOMER → ORDER
3. SUPERSTORE → BRANCH → EMPLOYEE

## **Task 5 – Reflection & Discussion**

**Find all customers across all branches who purchased a specific product type**

**Why is this type of query difficult in a hierarchical database?**

*Navigation Query*

SUPERSTORE → BRANCH → DEPARTMENT → ORDER → PRODUCT

This type of query is difficult in a hierarchical database since it's organized in a tree-like structure. It would need to traverse every branch under superstore and for each branch it looks at every customer. For each customer, it goes through all their orders to find the order which matches the target product type.

**What workarounds could be used to support it?**

Data Redundancy is a workaround to support this query. Every time a customer purchases a specific product type their information can be duplicated and stored at a higher level in hierarchy. For instance, the branch node can have a list of customers who purchased a specific product type.

**How would a relational database handle this more easily?**

A relational database would handle this query easily because it supports many to many relationships through junction tables and it returns all customers across all branches without manually traversing each branch.