

FOOD BANK

Course No., Section No.: 6360.003

Team Members:-

Name	Net ID
1)Rinkle Seth	rscs170004
2)Aditi Dixit	asd170007
3)Abhilasha Devkar	acd170130

1)PURPOSE:

- To store the data about food items which are supplied from different suppliers.
- To keep a record of food items which are delivered to zipcodes via agencies.
- To maintain a stock of food items stored in the inventory.
- To do a quality check to ensure that the items supplied from suppliers and delivered to agencies aren't expired.
- To maintain weekly updates about the food item requirements from agencies and plan future supply intakes accordingly.
- To keep a record of all employees, the manager of each section as well as the supervisor of employees if any.

2)DATA REQUIREMENTS:

- This is a food bank system database design for a single food bank organization which can take supplies from multiple suppliers(retailers, manufacturers, individuals, etc) and deliver it to multiple locations via multiple food agencies.
- One food agency can deliver food items to only one area/zip code.
- The food Items are stored in an inventory which has multiple sections classified according to the food type.
- The food bank can have many employees which work in different sections, every section has a manager and every employee has a supervisor.
- Suppliers will interact to employees for supplying food and also, food agencies can give orders of food items to employees.

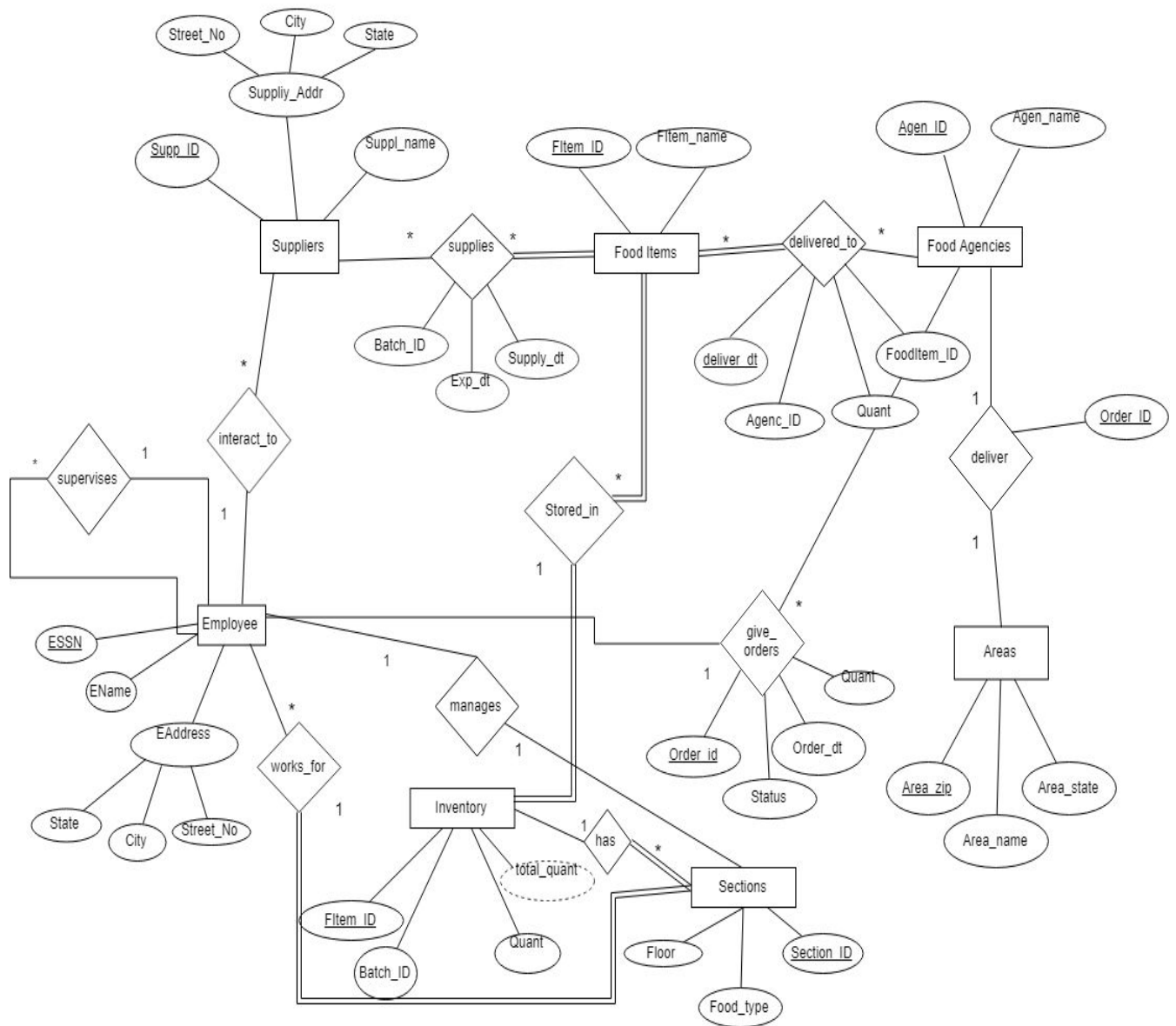
The data requirements can be translated into the following Entities and Relations.

Entities:- Suppliers, Food Agencies, Food Items, Employee, Areas, Inventory, Sections

Relations:- supplies, delivered_to, deliver, stored_in, interact_to, give_orders, manages, supervises, works_for.

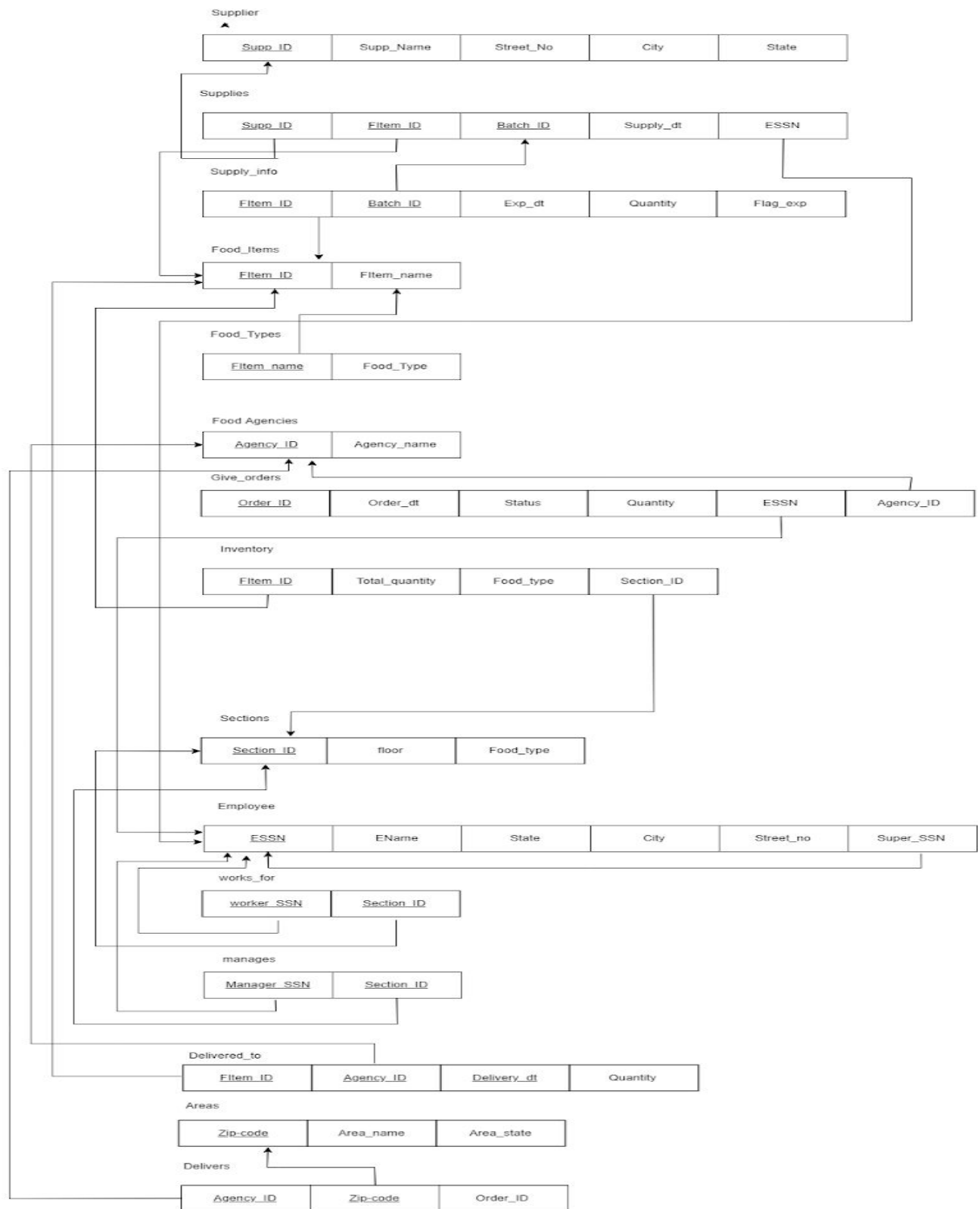
3)ER DIAGRAM:

Entity Relationship Diagram



4) RELATIONAL SCHEMA:

ER-Relational Mapping



:-

5)PROCEDURES:

1)PROCEDURE FOR EXPIRY CHECK

This is a procedure to check if the food items in the inventory have expired. It will compare the expiry date of the food item corresponding to the Batch Id with today's date and set the expired flag to true if the date has been crossed.

```
CREATE OR REPLACE PROCEDURE Expiry_Check()
is
DECLARE
cursor expiry
is
select Fitem_Id, B_id, exp_dt from Supply_Info;
Food_Item Supply_Info.Fitem_Id%TYPE
b_id Supply_Info.Batch_ID%TYPE
Exp_date Supply_Info.exp_dt%TYPE
begin
OPEN expiry
loop
fetch expiry into Food_Item, b_id,Exp_date ;
exit when expiry%NOTFOUND;
if(Exp_date<Sys.date):
    Update Flag_Expire
    set flag_exp = true
    where Fitem_Id = FoodItem and Batch_ID = B_Id;
    DBMS_OUTPUT.put_line("Some items from the
inventory have expired.");
End
```

II) CALCULATE WEEKLY_ORDERS

This procedure is designed in order to inform the managers about the records in the earlier week so as to help them manage the in-takes for the food items in the for the next weeks.

This procedure calculates the total number of orders received from 7 days before today's date a to today. These orders are segregated based on the areas of demand using zip codes.

```
CREATE OR REPLACE PROCEDURE Weekly_Orders()  
is  
DECLARE  
cursor orders  
is  
    select count(order_id), zip_code  
    from Give_Orders, delivers  
    where Give_orders.Order_Id = Del ivers.Order_Id  
    and  
        order_dt>(sys.date-7) and order_dt<=sys.date  
    group by ZipCode;  
count_orders number(15)  
Zip Delivers.ZipCode%Type;  
begin  
OPEN  
loop  
fetch orders into count_orders, Zip;  
exit when orders%NOTFOUND;  
DBMS_OUTPUT.put_line("The number of orders from area  
"+Zip+"for this week is:"+count_orders);  
end
```

6)TRIGGERS:

I)STOCK UPDATE TRIGGER AFTER SUPPLY

The food inventory is notified about all the supplies via this trigger. It executes every time a new supply is inserted into the database. It does so by incrementing the total_quantity record by the quantity in the newly added supply record.

```
CREATE OR REPLACE TRIGGER Stock_Update
    AFTER INSERT ON Supplies
FOR EACH ROW
WHEN( supplies.supp_ID >0)
BEGIN
    UPDATE Inventory i
    set i.total_Quant = new.quantity+I.total_Quant
    where i.FItem_Id = new.FID
END;
```

II) STOCK UPDATE AFTER DELIVERY

Whenever a particular quantity of a food Item has been delivered, it should reflect in the inventory as well. So, this trigger is invoked when a new record to delivered_to Relation is added.

This trigger reads the quantity from this record and subtracts it from the total Quantity of that food Item in the inventory.

```
CREATE OR REPLACE TRIGGER Stock_Update2
    AFTER INSERT ON delivered_to
FOR EACH ROW
WHEN( Fitem_ID >0 and agency_ID >0 and delivery_dt>0)
BEGIN
    UPDATE Inventory i
    set i.total_Quant = I.total_Quant - new.Quantity
    where i.FItem_Id = new.FID
END;
```