HYGIENIC FOOD DISTRIBUTION DATABASE

PHASE-3 Relational mapping and functional dependencies

We convert the ERR diagram to relational mapping with the help of the min max where we pull the entities from one to the other and do the relational mapping between them

We do the following steps in mapping they are:

- ✓ Identifying both primary keys and foreign keys among all the relations, which can be seen in terms of arrows from one entity to the other entity.
- ✓ Identify the functional dependencies for all the relations and show them in a table
- ✓ Identify all the candidate keys in the relations

Primary key:

A primary key is a special column in the relational database table which has unique value for every row element which distinguishes all the row elements in the column. A table can have only one primary key whereas it cannot be a null value.

Foreign Key:

A foreign key is used to connect the tables in the relational database. Primary key of table A is taken as foreign key of table B to make a relation between them, A table can have multiple foreign keys and there can be null values in the foreign keys.

Candidate Key:

A candidate key can also be called as the minimal super key as it is subset of the super keys, all the candidate keys are unique and not null values, every table can have at least one candidate key.

Functional dependency:

Functional dependency is a constraint which determines the relation between one attribute to the other in a database, plays vital role how good the database is and it is denoted by an arrow (→)

Functional dependency between A and B can be denoted as A → B

Primary keys in the database:

The following are the list of primary keys that are associated with each entity they are

Entity	Primary key
Employee	Employee Id
Customer	Customer Id
Kitchen	Kitchen Id
Order	Order Id
Discount	Discount Id
Menu	Food Id
Raw material	Distributor Id
Region	Region Id

Foreign Keys in the database:

Entity	Foreign Key
Order	Delivery person Id
Order	Kitchen Id
Order	Customer Id
Order	Food Id
Kitchen	Manager Id
Kitchen	Region manager Id
Customer	Discount Id
Purchase raw materials	Distributor Id
Purchase raw materials	Kitchen Id
Works for kitchen	Employee Id
Works for kitchen	Kitchen Id
Displays menu	Kitchen Id
Displays menu	Food Id

Candidate Keys:

Employee: Employee Id, Email, Phone number, SSN

Customer: Customer Id, Phone number, Email

Kitchen: Kitchen Id, Email, Phone number

Order: Order Id

Discount: Discount Id

Menu: Food Id

Raw material: Distributor Id

Region: Region Id, Regional manager Id, Region Name

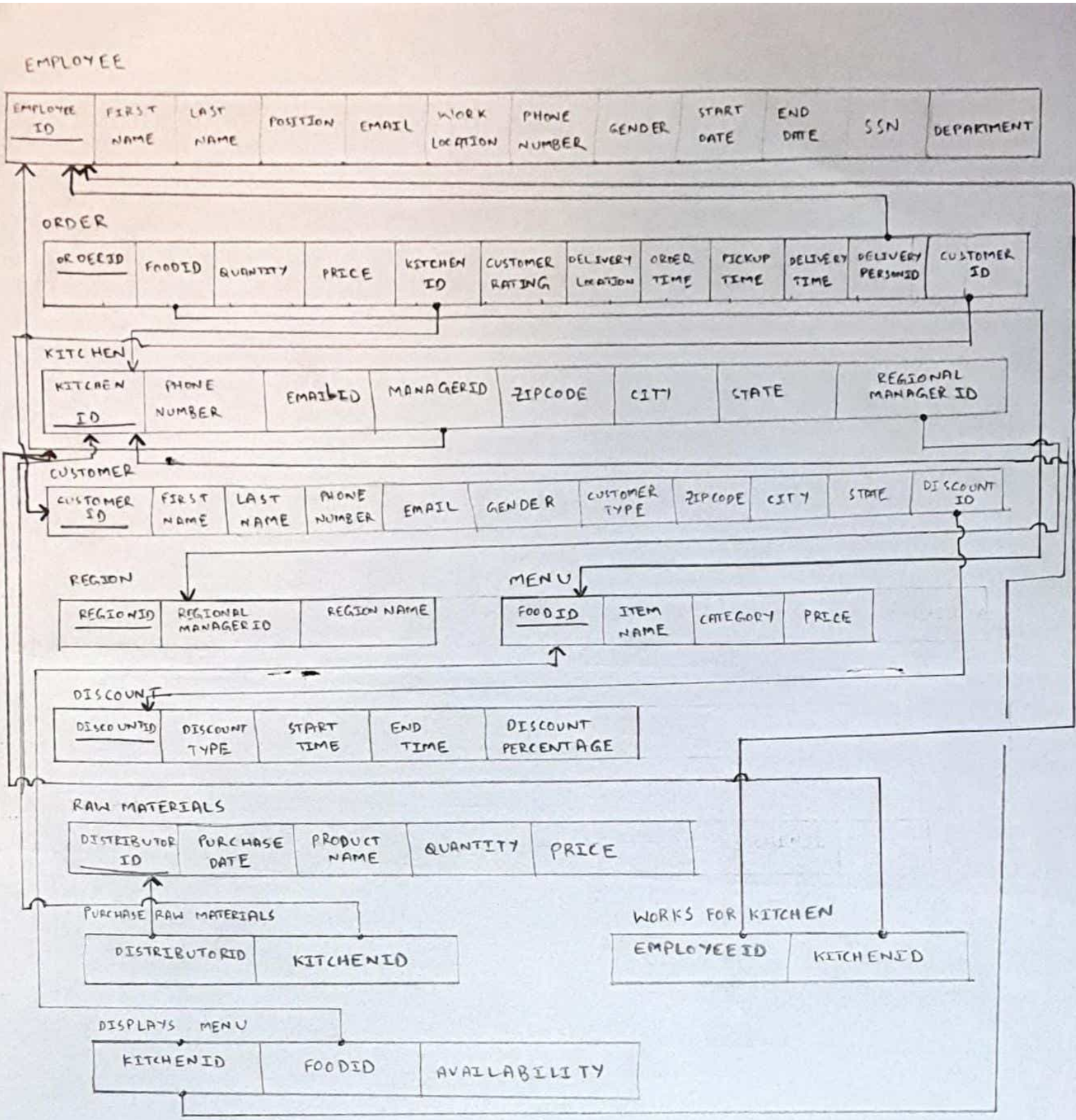
Purchases Raw material: Distributor Id, Kitchen Id

Works for Kitchen: Employee Id, Kitchen Id

Displays Menu: Kitchen Id, Food Id

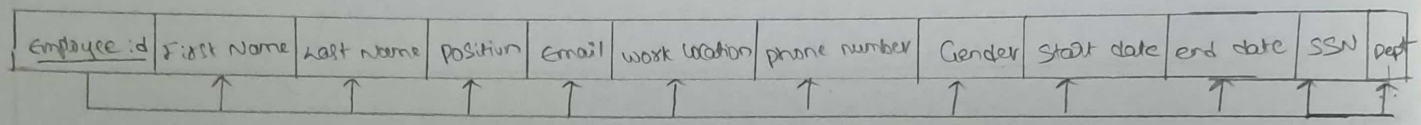
Assumptions on the relational mapping:

- The relation between Employee and Order is ‘delivers’ which has (0, N) towards the employee and (1,1) towards Order. So, for the relational mapping we pulled Employee key (delivery person Id) towards the Order.
- One of the relations between Employee and Kitchen is ‘works for’ which has (1, N) towards the employee and (4, N) towards Kitchen. So, we can’t pull from either entity so we create new relation which is works for which the primary key will be combination of employee Id and Kitchen Id.
- One of the relations between Employee and Kitchen is ‘manages’ which has (0,1) towards the employee and (1,1) towards the kitchen. So, for the relational mapping we pull employee key (manger Id) towards the Kitchen.
- The relation between Orders and Customer is ‘places order’ which has (0,N) towards the customer and (1,1) towards the order. So, for the relational mapping we pull customer key (customer id) towards the Order.
- The relation between kitchen and Menu is ‘Displays’ which has (1,N) towards the menu and (1,N) towards the Kitchen. So, for the relational mapping we can pull either side which gives us new relation which is Displays for which the primary key will be combination of Food Id and Kitchen Id. Additionally it has an attribute as availability used as a flag to manages the display items in the menu.
- The relation between the region and kitchen is ‘manages’ which has (1, N) towards the region Id and (1,1) towards the Kitchen. So, for the relational mapping we pull the region key(region manger id) towards the Kitchen.
- The relation between the Raw material and Kitchen ‘Purchases’ which has (0,N) towards the Raw materials and (1,N) towards the Kitchen. So, relational mapping is not possible on the either side so we create a new relation which is ‘Purchases Raw materials’ which has Distributor Id and Kitchen Id as the primary key.
- The relation between Discount and customer is ‘Gets discount’ which has (1,N) towards Discount and (0,1) towards customer. So, for the relational mapping we pull the Discount key value(discount id) towards the customer as it has lesser null values compared to the other.

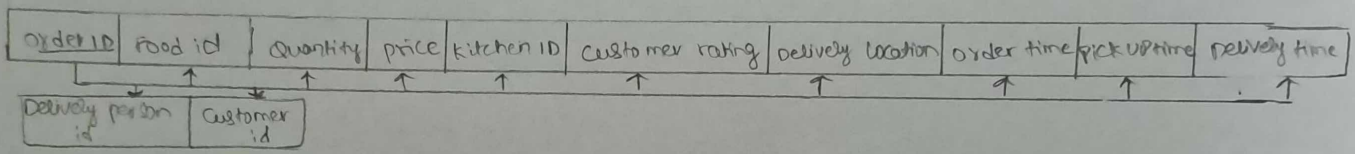

EMPLOYEE
EMPLOYEE ID
FIRST NAME
LAST NAME
POSITION
EMAIL
WORK LOCATION
PHONE NUMBER
GENDER
START DATE
END DATE
SSN
DEPARTMENT
ORDER
ORDERID
FOODID
QUANTITY
PRICE
KITCHEN ID
CUSTOMER RATING
DELIVERY LOCATION
ORDER TIME
PICKUP TIME
DELIVERY TIME
DELIVERY PERSONID
CUSTOMER ID
KITCHEN
KITCHEN ID
PHONE NUMBER
EMAILID
MANAGERID
ZIPCODE
CITY
STATE
REGIONAL MANAGER ID
CUSTOMER
CUSTOMER ID
FIRST NAME
LAST NAME
PHONE NUMBER
EMAIL
GENDER
CUSTOMER TYPE
ZIPCODE
CITY
STATE
DISCOUNT ID
REGION
REGIONID
REGIONAL MANAGERID
REGION NAME
MENU
FOODID
ITEM NAME
CATEGORY
PRICE
DISCOUNT
DISCOUNTID
DISCOUNT TYPE
START TIME
END TIME
DISCOUNT PERCENTAGE
RAW MATERIALS
DISTRIBUTOR ID
PURCHASE DATE
PRODUCT NAME
QUANTITY
PRICE
PURCHASE RAW MATERIALS
DISTRIBUTORID
KITCHENID
WORKS FOR KITCHEN
EMPLOYEEID
KITCHENID
DISPLAYS MENU
KITCHENID
FOODID
AVAILABILITY

Functional dependency:

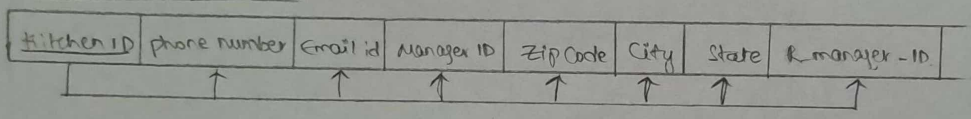
Employee



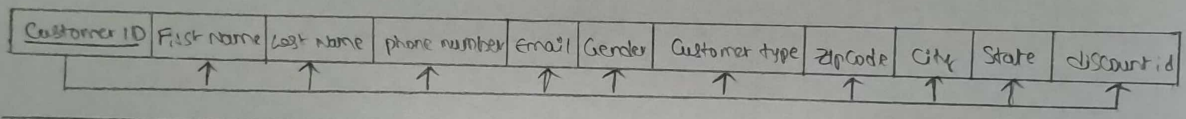
Order



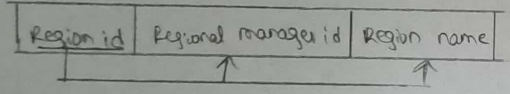
kitchen



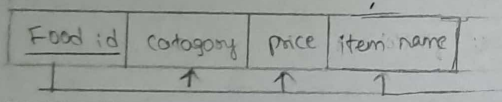
Customer



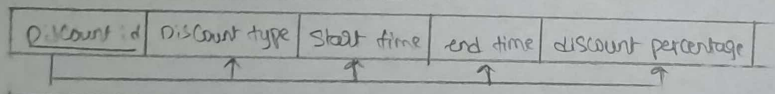
Region



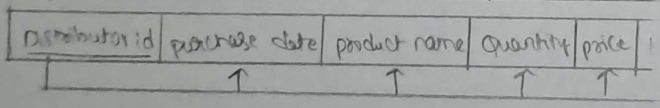
Menu



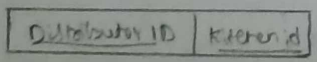
Discount



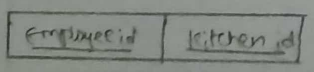
Raw materials



purchases Raw materials



works for Kitchen



Displays Menu