SSN COLLEGE OF ENGINEERING, KALAVAKKAM, CHENNAI

DBMS THEORY ASSIGNMENT – 1

FOOD DELIVERY MANAGEMENT SYSTEM

DONE BY:

Kushaal Shyam Potta 3122235001071

Mahadev Ramesh Ramya 3122235001075

Murari Sreekumar 3122235001087

Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam – 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)
Computer Science and Engineering
UCS2404 - DATABASE MANAGEMENT SYSTEMS
Regulations – R2021

<table>
<tr><td>Course Code</td><td>UCS2404</td><td>Course Name</td><td colspan="6">Database Management Systems</td></tr>
<tr><td rowspan="2">Course Type</td><td rowspan="2">Theory</td><td rowspan="2">Course Category</td><td rowspan="2">Professional Core (PC)</td><td>L</td><td>T</td><td>P</td><td>E</td><td>C</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td>Regulation</td><td colspan="2">R-2021</td><td>Academic Year</td><td colspan="5">2024-25 (Even)</td></tr>
<tr><td>Degree and Branch</td><td colspan="2">B.E. Computer Science & Engineering</td><td>Batch</td><td colspan="5">2023-27</td></tr>
<tr><td colspan="2">Semester</td><td>IV</td><td>Faculty Name</td><td colspan="5">Dr. P. Mirunalini
Dr. N. Sujaudeen</td></tr>
<tr><td colspan="3">Department Offering the Course</td><td colspan="6">Computer Science and Engineering</td></tr>
</table>

QUESTION:

a. Phase-1: Identification of constraints and Functional dependencies (FD) among set of attributes

1. Analyze the data for a specific real-time application which uses database system as listed in the following:

   a. Library Management System

   b. Gas supply chain Management system

   c. Hospital Management System

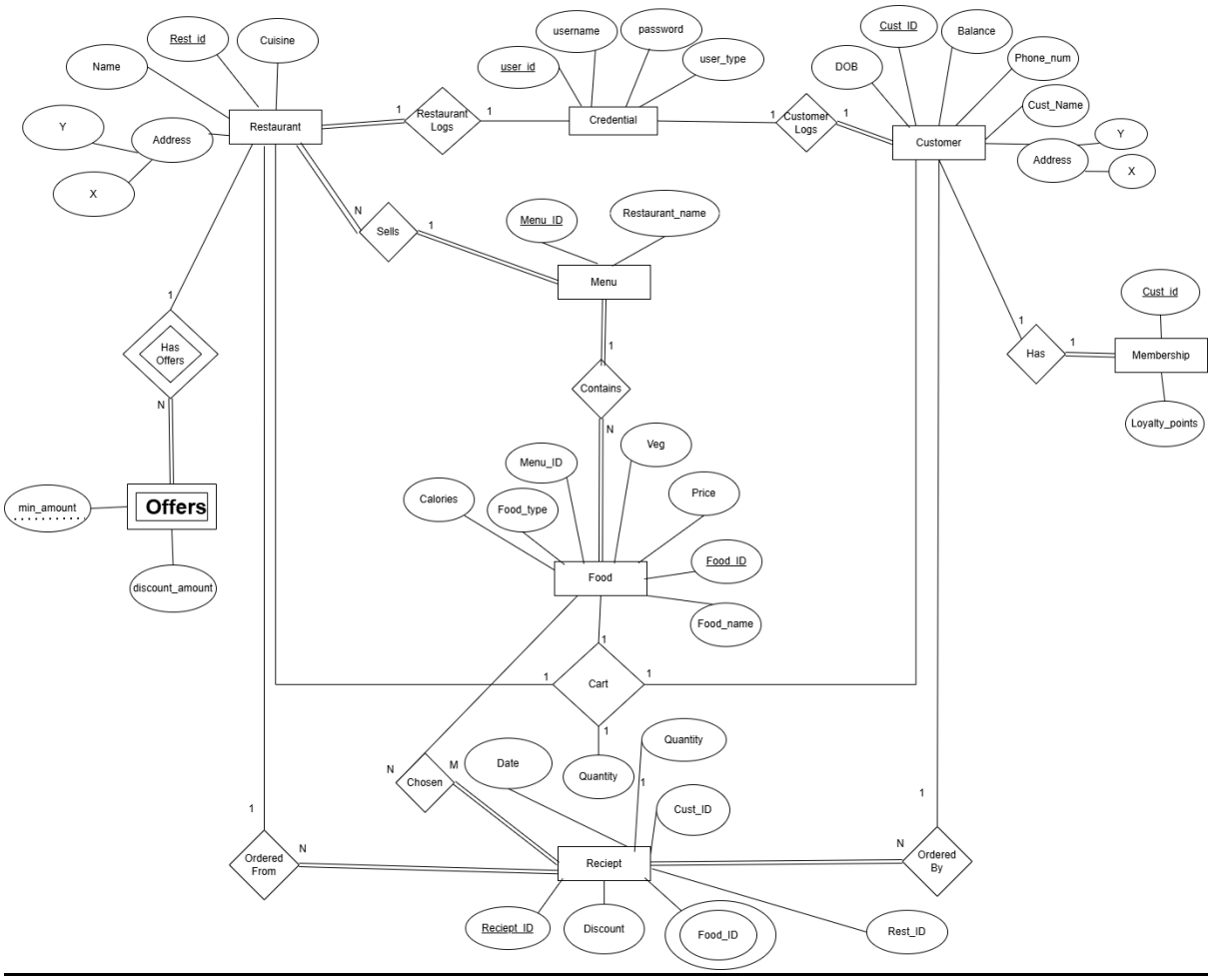
   d. Food Delivery Management system

   e. Journal/Conference Publication management system.

Analyzing the data includes identifying the constraints among the attributes and the existence of dependencies among the attributes. The constraints and dependencies identification could be framed/identified by deeper understanding of the problem specification, interactions/dependencies among various attributes and user-requirements for the specific real-time application. **[PO2-2.1.2, 2.1.3, 2.3.2, 2.4.1]**

2. Prepare the documentation showing the list of attributes and identify various functional dependencies among the set of attributes. **[PO10-10.1.1]**

ENTITY-RELATION DIAGRAM:



ENTITIES AND THEIR ATTRIBUTE TYPES:

Restaurant

1) Restid:Key attribute (varchar2)
2) Name(singlevalued,stored,simple) (varchar2)
3) cuisine(singlevalued,stored,simple) (varchar2)
4) Address: (Composite)
    - X:Number
    - Y:Number

Credential

1) User_id:key attribitue (Varchar2)

2) Username (Varchar2) (simple,single valued,stored attribute)
3) Password (Varchar2) (simple,single valued)
4) Usertype (Varchar2) (simple,single valued,stored attribute

Customer

1) Custid: (Varchar2) (key attribute)
2) Address:(composite)
    - X: (Number)
    - Y: (Number)
3) Cust_name: (Varchar2) (simple,single valued,stored attribute)
4) Dob: (Date) (simple,single valued,stored attribute)
5) phonenum:(Number)(Single valued,Simple)
6) Balance :(Number) (simple,singlevalued)

Menu

1) Menuid:(Varchar2) Key Attribute
2) Restaurant_name(Varchar2): (simple, singlevalued)

Membership

1) Cust_id:(key attribute) (Varchar2)
2) Loyalty points :Number (simple,single)

Food

1) Food_id:String (key attribute) (Varchar2)
2) Food_name: (Varchar2) (Simple,single-valued)
3) Calories: (Number)(Simple,single-valued)
4) Food_type: (Varchar2) (Simple, single-valued,Stored)
5) Menu_ID: (Varchar2) (simple,single valued,stored)
6) Veg: (Varchar2) (Simple, single-valued,stored)
7) Rating: (Number) (Simple,single_valued)
8) Price: (Number) (Simple,single-valued)
1))

Receipt

1) Receiptid:key attribute (Varchar2)

2)Discount: (simple,single valued,stored),(number)

3) quantity:(simple,single valued,stored),(number)

4) custid:(simple,single valued,stored),(varchar2)

5) restid:(simple,single valued,stored),(varchar2)

6) foodid:(multivalued),(varchar2)

Offers:

1) Min_amount: (Partial-key) (Number)
2) Discount_amount: (Number) (Simple, single-valued)

ER-Diagram to Relation Schema conversion:

Credentials:

User_ID (Primary-Key)

Username (Simple-Attribute)

Password (Simple Attribute)

User_type (Simple Attribute)

Restaurant:

Rest_ID (Primary Key) (Refers to Credentials (User_ID))

Cuisine (Simple-Attribute)

Name (Simple-Attribute)

X (Simple-Attribute)

Y (Simple-Attribute)

Menu (Foreign-Key)

Since there exists a 1:1 relationship between credentials and restaurant with restaurant in total participation and credential as partial, we have to add User_ID (Credential) to the restaurant's schema, however as the Rest_ID already refers to it, we do not add it here.

As restaurant shares a N:1 relation 'sells' with Menu, we add Menu's primary key Menu_ID as a foreign key.

Customer:

Cust_ID (Primary key)

Balance (Simple-Attribute)

Cust_name (Simple-Attribute)

Phone_num (Simple-Attribute)

Dob (Simple-Attribute)

X (Simple-Attribute)

Y (Simple-Attribute)

Since there exists a 1:1 relationship between credentials and customer with customer in total participation and credential as partial, we have to add User_ID (Credential) to the customer's schema, however as the Cust_ID already refers to it, we do not add it here.

Menu:

Menu_ID (Primary Key)

Restaurant_name (Foreign key refers Restaurant(Name))

Offers:

Rest_ID, Min_amount (Composite Key) (Rest_ID refers to Restaurant (Rest_ID))

Discount_amount (Simple attribute)

Since Offers is in a weak entity relationship with Restaurant, the Rest_ID attribute is added into offers referencing Rest_ID in restaurant and Rest_ID and partial key Min_amount behaves as a composite key.

Membership:

Cust_ID(Primary key refers Customer (Cust_ID))

Loayalty_points (Simple Attribute)

Since membership has a 1:1 relationship 'Has' with customer with total participation from 'Membership' and partial participation from 'Customer', we need to add Cust_ID as a foreign key attribute here, however as the primary key already refers to it, we don't add it.

Food:

Food_ID (Primary Key)

Price (Simple attribute)

Veg (Simple attribute)

Food_name (Simple attribute)

Food_type (Simple attribute)

Calories (Simple attribute)

Menu_ID (Foreign key refers Menu(Menu_ID))

Food shares a N:1 relationship ‘Contains’ with Menu, so we add Menu’s primary key, Menu_ID as a foreign key here.

Cart:

Restaurant_ID,Food_ID,Customer_ID (Composite Key refers Restaurant, Food and customer respectively)

Quantity (Simple attribute)

As Cart is a ternary relationship between Restauarnt,Food and Customer it will take their respective primary keys as a composite key along with the additional attribute quantity.

Receipt:

Receipt_ID (Primary Key)

Date (Simple attribute)

Quantity (Simple attribute)

Rest_ID (Foreign key refers to restaurant (Rest_ID))

Cust_ID (Foreign Key refers to customer (Customer_ID))

Receipt shares a N:1 relationship ‘Ordered by’ with Customer, we add customer’s primary key as a foreign key here.

Receipt shares a N:1 relationship ‘Ordered from’ with Restaurant, we add restaurant’s primary key as a foreign key here.

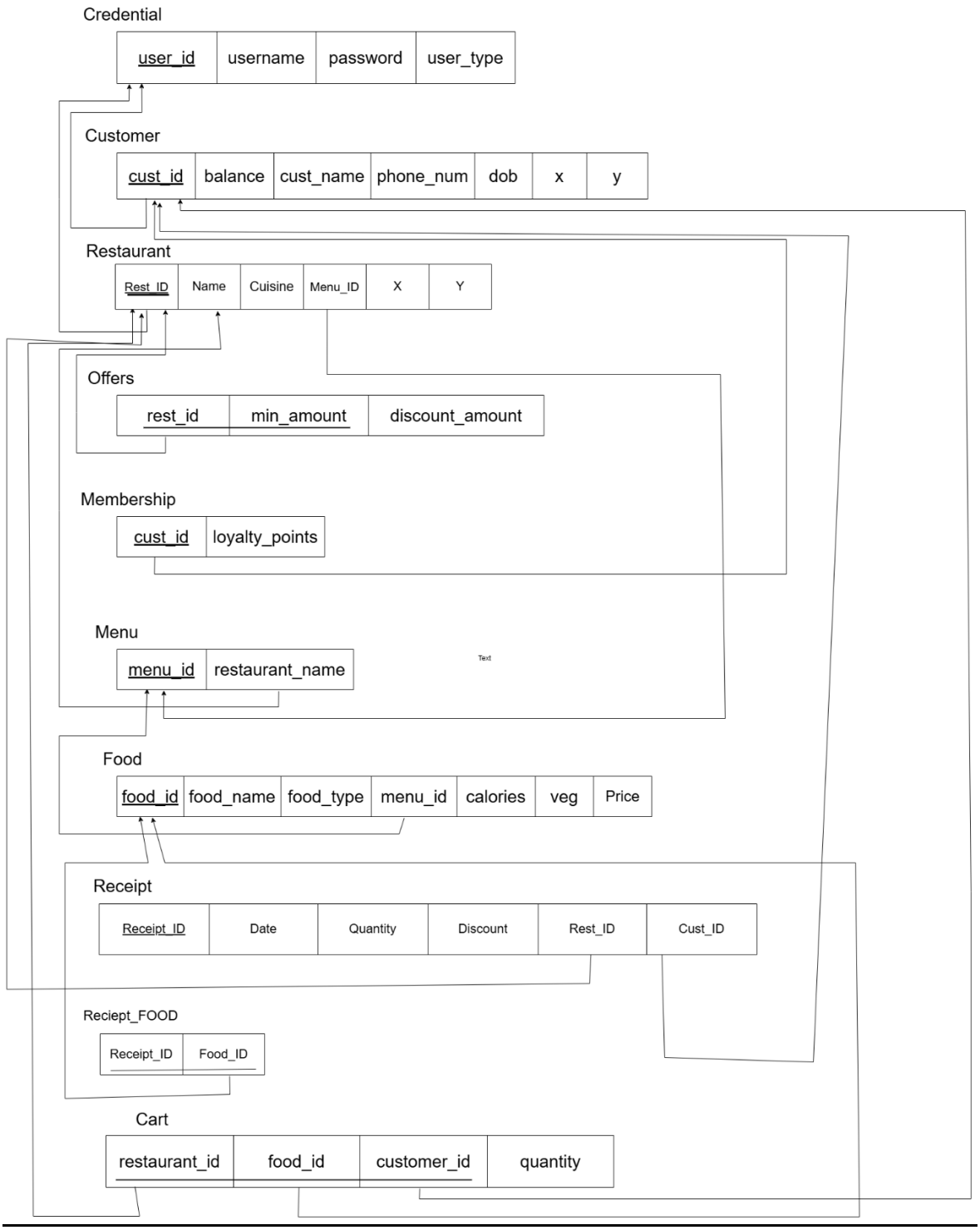
Receipt-Food:

Reciept_ID,Food_ID (Composite key refers Restaurant and Food)

As Food_ID is a multi-valued attribute in Receipt entity, we create a separate relation with Reciept’s primary key, Receipt_ID and Food_ID as composite key.

Receipt also shares a N:M relationship ‘Chosen’ with food, for which too this table satisfies the relation required.

SCHEMA DIAGRAM (ER TO RELATIONAL MAPPING):



FUNCTIONAL DEPENDENCIES:

Relation 1:Credentials

List of FD's:

user_id -> username

user_id -> password

user_id -> user_type

Minimal set of FD's:

user_id -> username

user_id -> password

user_id -> user_type

Relation 2: Customer

List of Functional Dependencies:

cust_id -> balance

cust_id -> cust_name

cust_id -> phone_num

cust_id -> dob

cust_id -> x

cust_id -> y

phone_num -> cust_id

phone_num -> balance

phone_num -> cust_name

phone_num -> dob

phone_num -> x

phone_num -> y

Removal of FD's:

We remove the FD's as follows: cust_id -> balance, cust_id -> cust_name, cust_id -> dob, cust_id -> x, cust_id -> y, since they are implied through cust_id -> phone_num and the other FD's of phone_num through transitive property

Set of Minimal FD's:

cust_id -> phone_num

phone_num -> cust_id

phone_num -> balance

phone_num -> cust_name

phone_num -> dob

phone_num -> x

phone_num -> y

Relation 3: Restaurant

List of Functional Dependencies:

rest_id -> name

rest_id -> cuisine

rest_id -> x

rest_id -> y

rest_id -> menu_id

name -> menu_id

name -> cuisine

menu_id -> name

Menu_id -> cuisine

x,y -> rest_id

x,y -> menu_id

Removal of FD's:

- We remove the FD's rest_id -> cuisine, rest_id -> menu_id since they are implied from the FD's name -> cuisine, name -> menu_id through transitivity property.
  .

Set of Minimal FD's:

rest_id -> name

rest_id -> x

rest_id -> y

name -> menu_id

Menu_id -> cuisine

menu_id -> name

x,y -> rest_id

x,y -> menu_id

Relation 4: Offers

List of Functional Dependencies:

rest_id, min_amount -> discount_amount

Relation 5: Membership

List of Functional Dependencies:

cust_id-> loyalty_points

Relation 6: Menu

List of Functional Dependencies:

menu_id -> restaurant_name

restaurant_name -> menu_id

Relation 7: Food

List of Functional Dependencies:

food_id -> food_name

food_id -> food_type

food_id -> menu_id

food_id -> calories

food_id -> veg

food_id -> price

food_name -> food_type

food_name -> veg

food_name, menu_id -> calories

food_name, menu_id -> price

food_name, menu_id -> food_id

food_name, menu_id -> veg

Removal of FD's:

- We remove the FD's food_id -> food_type, food_id -> veg since they are implied using the FD's food_id -> food_name, and food_name -> food_type and food_name -> veg through the transitive property.
- Upon removal of menu_id in the FD food_name, menu_id -> veg, we get food_name -> veg, which already exists, it is enough to use food_name -> veg represent the FD (food_name, menu_id -> veg), hence we remove food_name, menu_id -> veg.

Set of Minimal FD's:

food_id -> food_name

food_id -> menu_id

food_id -> calories

food_id -> price

food_name -> food_type

food_name -> veg

food_name, menu_id -> calories

food_name, menu_id -> price

food_name, menu_id -> food_id

Relation 8: Receipt

List of Functional Dependencies:

Receipt_ID -> Date

Receipt_ID -> Restaurant_ID

Receipt_ID -> Customer_ID

Receipt_ID -> Quantity

Receipt_ID -> Discount

Minimal list of FD's:

Receipt_ID -> Date

Receipt_ID -> Restaurant_ID

Receipt_ID -> Customer_ID

Receipt_ID -> Quantity

Receipt_ID -> Discount

Relation 9: Reciept_Food

There are no functional dependencies in this relation.

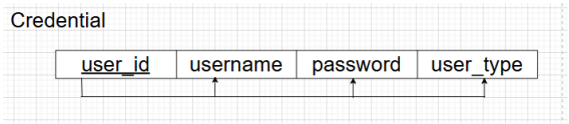
Relation 10: Cart

List of Functional Dependencies:

restaurant_id, food_id, customer_id -> quantity

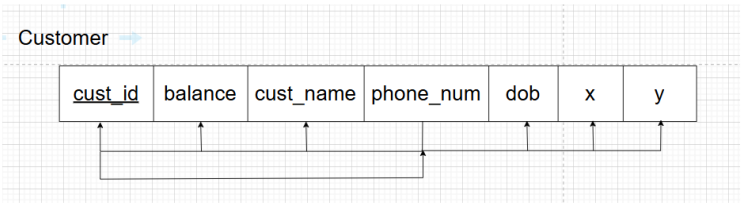
Normalisation:

Normalization: Relation 1 - Credential



This following relation is already in 1NF since all attributes are atomic, is in 2NF since there is no partial dependency, It is also in 3NF since it has no transitive dependencies. It is also in BCNF

Functional Dependencies: Relation 2 - Customer



Minimal set of FD's:

cust_id -> phone_num

phone_num -> cust_id

phone_num -> balance

phone_num -> cust_name

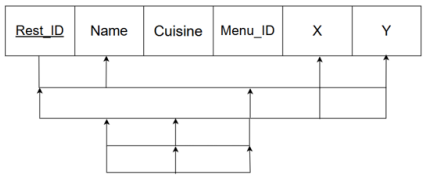
phone_num -> dob

phone_num -> x

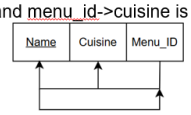
phone_num -> y

Normalization: Relation 3 - Restaurant

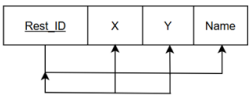
Restaurant



The above relation is in 1NF since all attributes are atomic, it is in 2NF since there are no partial dependencies, it is NOT in 3NF because of the FD's rest_id -> name, name -> menu_id where name is NOT a super key/candidate key. Therefore, we decompose the table. Even name->menu_id and menu_id->cuisine is also transitive dependency as menu_id is not a candidate key.



The first relation is in 3NF . It is also in BCNF as name,menu_id are superkeys.



The second relation is in 3NF since there are no transitive functional dependencies. It is also in BCNF. As x,y and rest_id are superkeys

Minimal set of FD's:

rest_id -> name

rest_id -> x

rest_id -> y

name -> menu_id

Menu_id -> cuisine

menu_id -> name

x,y -> rest_id

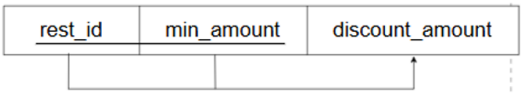
x,y -> menu_id

Normalization: Relation 4 - Offer

List of FD's:

rest_id, min_amount -> discount_amount

Offers



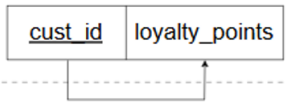
This following relation is already in 1NF since all attributes are atomic, is in 2NF since there is no partial dependency, It is also in 3NF since it has no transitive dependencies, It is in BCNF because the determinant part of the FD's are superkeys.

Normalization: Relation 5 - Membership

List of FD's:

cust_id-> loyalty_points

Membership



This following relation is already in 1NF since all attributes are atomic, is in 2NF since there is no partial dependency, It is also in 3NF since it has no transitive dependencies, It is in BCNF because the determinant part of the FD's are superkeys.

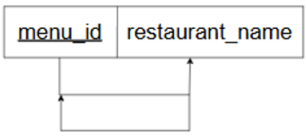
Functional Dependencies: Relation 6 - Menu

List of FD's:

menu_id -> restaurant_name

restaurant_name -> menu_id

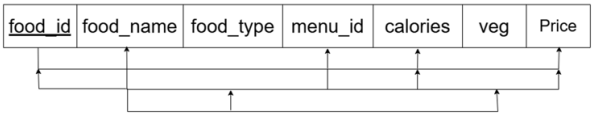
Menu



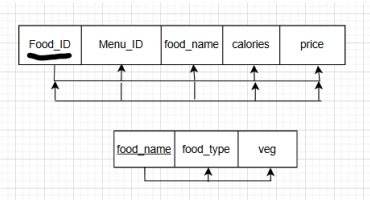
This following relation is already in 1NF since all attributes are atomic, is in 2NF since there is no partial dependency, It is also in 3NF since it has no transitive dependencies, It is in BCNF because the determinant part of the FD's are superkeys.

Normalization: Relation 7 - Food

Food



The above relation is in 1NF since all attributes are atomic, it is in 2NF since there are no partial dependencies, it is not in 3NF because of the transitive dependency food_id -> food_name, food_name -> food_type, food_name -> veg where food_name is NOT a candidate key/super key. Therefore, we decompose the table.



First relation is in 3NF since there is no transitive dependency, it is also in BNCF as foodid and food_name.menu_id are superkeys.

Second relation is in 3NF since there is no transitive dependency. It is also in BCNF because food_name is a superkey.

Minimal set of FD's:

food_id -> food_name

food_id -> menu_id

food_id -> calories

food_id -> price

food_name -> food_type

food_name -> veg

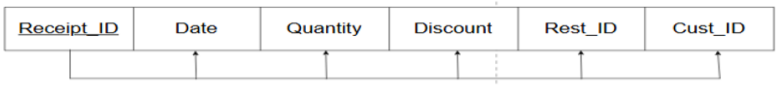
food_name, menu_id -> calories

food_name, menu_id -> price

food_name, menu_id -> food_id

Normalization: Relation 8 - Receipt

Receipt



Minimal list of FD's:

Receipt_ID -> Date

Receipt_ID -> Restaurant_ID

Receipt_ID -> Customer_ID

Receipt_ID -> Quantity

Receipt_ID -> Discount

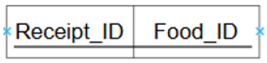
This following relation is already in 1NF since all attributes are atomic, is in 2NF since there is no partial dependency, It is also in 3NF since it has no transitive dependencies, It is in BCNF because the determinant part of the FD's are superkeys.

Normalization: Relation 9 - Receipt_food

List of FD's:

There are no functional dependencies in this relation.

Receipt_FOOD



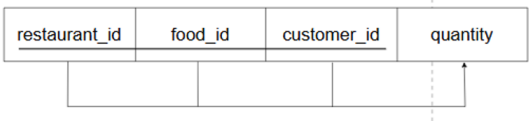
This following relation is already in 1NF since all attributes are atomic, is in 2NF since there is no partial dependency, It is also in 3NF since it has no transitive dependencies, It is in BCNF because the determinant part of the FD's are superkeys.

Normalization: Relation 10 - Cart

List of FD's:

restaurant_id, food_id, customer_id -> quantity

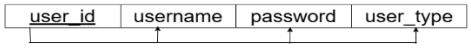
Cart



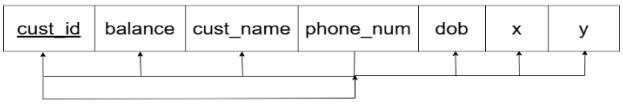
This following relation is already in 1NF since all attributes are atomic, is in 2NF since there is no partial dependency, It is also in 3NF since it has no transitive dependencies, It is in BCNF because the determinant part of the FD's are superkeys.

Final Schema (post normalisation)

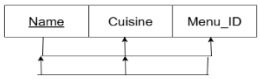
Credential

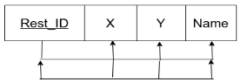


Customer

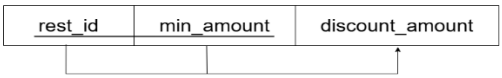


Restaurant



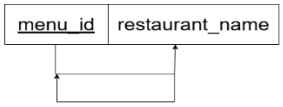


Offers

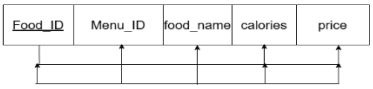


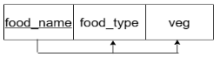
Membership

Menu

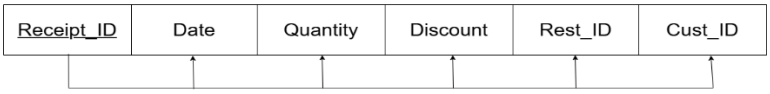


Food

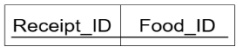




Receipt



Receipt_FOOD



Cart