Feature 1:
Input: customer name, address, state, zip, email
Output :First we need to check the input emailid with the existing records in the customer table If it exists then print "the client already exists" and also update the same record with new input address,state,zip.If It does not exist then update the table with new record with the input record by generating the id with sequence generation(Id) and credit as zero value and print that updated record.
Test case 1:

- ★ **Input**: customer name, address, state, zip, email
- ★ exec add_customer('Emily', '1345WestLane', 'MD',21042,'emily@gmail.com');

Output:'The client does not exist' and also print the newly added record.
Test case 2:

- ★ Input: customer name, address, state, zip, email
- ★ exec add_customer('John', '213Pratt', 'MA',27655,'john@gmail.com');

Output : print 'the client already exists' and updated record for this mail id.

Feature 2:
Input: email
Output :First we need to check the input emailid with the existing records(emails) in the customer table.If it exists then print the record for that mailid with the details customer name, address, state, zip code, email, credit, total number of orders with status 2 (delivered)in the last six months and total amount spent (sum of total cost for orders with status 2) in the last six months. We need to fetch these details by joining the customer and orders.If It does not exist then print 'no such customer'
Test case 1:

- ★ Input:emailid
- ★ exec new_customer_record('emily@gmail.com');

Output:'No such customer'
Test case 2:

- ★ **Input:**emailid
- ★ exec add_customer('john@gmail.com');

Output : print customername='John' address ='213Pratt', state ='MA', zipcode =27655, email='john@gmail.com', credit=26, total number of orders with status delivered in the last six months =1,total amount spent in the last six months=1000

Feature 3:
Input: Category name
Output: If the category name is existed in the table then we will print outputs the name of the restaurant, average review score, average wait time,and zip code
for restaurants that are open and match the category name.
If the category name doesn't exist in the table then the output would be 'invalid'.

Test Case 1 :

- **Input :** Sea
- Exec **(**category name = '%Sea%')
- As the categoryname is valid

Output: Restaurant name= paradise, average review score= 2, average wait time= 9.2 and zip code = 70731

Test Case 2 :

- **Input :** fast
- Exec (category name = '%Fast%')
- As the categoryname is invalid

Output : Invalid

<u>Feature 4</u>:
Input: Restaurant ID
Output: Checks if the restaurant ID is valid .If yes then all dishes in this restaurant, along with dish name and price are printed as output.
If the restaurant ID is not valid,a message 'no such restaurant' will be printed.
Test case 1:
-- Select dish name, price
From food items
Where restaurant ID = '22';
Exec
when no_data_found then
Dbms_output.put_line('Pasta',25)**;**

Test case 2:
-- Select dish name, price
From food items
Where restaurantid = 'given id';
Exec
when no_data_found then
Dbms_output.put_line('no such restaurant');

Member3:
<u>Feature 5:</u>
Input:=Cart id
Output:= First we need to check whether that cartID is valid. If not, print a message invalid cart ID. If the ID is valid, print out dish name, price, quantity.
Case 1:

- Input Cart id=6;
- Exec check (Cartid)=6;
- For the given cart id exists,So the dish name is noodles ,price is 20 and quantity of noodles is 10.

Case 2:

- Input Cart id=11;
- Exec check (Cartid)=11;
- For the given card id does not exists ,so we will get an error message with “Invalid cart id”.

Feature 6:

Input:=Dish id,Cart id

Output:=First we need to check whether the cart with the given ID has that dish. If not print a message 'Invalid’.If the input ID is valid, check the quantity of that dish.

If it is more than one, then reduce the quantity of that dish from the cart dish table and print a message saying ‘quantity reduced’. If the quantity is one, delete that row from the cart and print out 'dish removed

Case1:

- Input Dishid=46;
- Exec check (Dishid)=46;
- Dish id does not exist in the dish table,So if the dish is invalid we will get an error message with “invalid dish id.”

Case2:

- Input Dish id=44;
- Exec check (Dishid)=44;I
- Dish id 44 exists in the dish tables. We'll update the table cartdish table’s quantity. Attribute with the dish id 44. Update cart dish set quantity = quantity-1 where dishid=44;

Case3:

- Input Dish id=44;
- Exec check (Dishid)=44;
- Since the quantity is only one.so if we subtract the remaining one quantity from the dish so we are left with zero.Hence,dish will be removed from the restaurant.

Member4:

Feature 7:

Input : OrderID , status , input time.

Output : If the orderID does not exist, then we will print a message saying invalid orderID. If the orderID exists, we will check the status of the order if the status is in progress then no additional action is required . But, if the status is delivered, we will insert a new row into the message table with message time as input time, and the message body saying 'Your order X has been delivered!' where X is the order ID. But, if the new status is canceled then we will update the status to cancel and insert a message into the message table for the corresponding customer , with message time as input time, and the message body saying 'Your order X has been canceled and refund issued!'. The next step is to add a record in the payment table with the new payment id, customer id, order id , input time, payment method which is the original payment record and as the negative of the total amount in the order.

Case 1 :

- Input : 331
- Exec updatestatus(331)
- As the ordered is invalid
- Output : Invalid orderID

Case 2:

- Input : 123
- Exec updatestatus(123)
- OrderID exists and status is delivered
- Output : Insert into message values (916 , 1 , timestamp '2022-10-2 08:05:00.00' , 'Your Order X is Delivered'); (X=123).

Case 3:

- Input : 789
- Exec updatestatus(789)
- OrderID exists and status is Canceled

Output : Insert into message values (917 , 3 , timestamp '2022-10-2 10:05:00.00' , ''Your order X has been canceled and refund issued!'); (X=789).

Insert into payments values (816 , 3 , timestamp '2022-10-2 08:05:00.00' , 789 , -70 , 'applepay');

Feature 8:

Input:= customer id, restaurant id, review date, review score, comment

Output:= First we will validate the customer id and restaurant id if any one of the both is invalid we will print "invalid customer id" or "invalid restaurant id".

If both the customer id and restaurant id are valid then we will populate the data into the table and update the avg review score in the restaurant table.

Case1:

- Input Customerid=6;
- Exec check(customerid)=6;
- Customer id does not exist in the customer table, So customer id is invalid we will get an error message with
  - *"Invalid customer id."*

Case2:

- Input restaurantid = 6
- Exec check(restaurantid)=6;
- Restaurant does not exist in the restaurant table, So restaurant id is invalid we will get an error message with
  - *"Invalid restaurant id"*

Case3:

- Input customerid = 2 restaurantid= 101, review date '12-JAN-20', review score = 3 and comment = 'good'.
- Check (inputs)
  - *As the customerid and restaurantid are valid inputs we populate the data with a given review date, review score and comment.*
  - Then we have to update the avg review score of that restaurant table with (2+3)/2 i.e **2.5.**

Feature 9:

Input: Restaurantid

Output: First validate Restaurant id if it exists in the restaurant table then we will fetch data from the review table and print restaurant name,reviewdate, review score, comment.

Case 1:

- Input Restaurantid =201
- Exec check(Restaurantid);
- Restaurantid 201 is not existed in the given table, So we will get an error message with :
  - *"Not a valid restaurant id".*

Case 2:

- Input restaurantid=101
- Exec check(restaurantid);
- Restaurantid "101" is existed in the given table, So we will get the restaurant name with all the review scores and comments.
  - Ex: restaurant name: **Persis** review date: 12-JAN-20 review score: 2 comment: good

Feature 10:

Input: customerid, restaurantid, dishid

Output: We will validate the inputs if they are valid then we update the cartdish table with the given inputs

Case 1:

- Input Customerid=6
- Exec check(customerid);
- Customer is not existed in the customer table so we will the get error message with
  - "*Not a valid customer id".*

Case 2:

- Input Restaurantid=6
- Exec check(restaurantid,status)
- Restaurant is not existed in the restaurant table so we will the get the error message
  - "Not a valid Restaurant id".

Case 3:

- Input Restaurant=102
- Exec(restaurantid,status)
- Restaurant is existed but the status was closed so we get a message with "restaurant is closed'.

Case 4:

- Input Restaurantid=101, customerid=1
- Exec check(Restaurantid, customerid,status);
  - The given restaurantid and customerid are valid and restaurant status is open , So we have to update the cart table
    Ex: Exec update(cart table) with cartid,customerid,restaurantid.

Feature 11:

Input : cartid, check out time, delivery method

Output : If cartid is not valid then print "Invalid cart id" else print the sum up price of the customer on his order for the items after calculating tax, discounts and delivery method.

Case 1:

- Input Cart id =11;
- Exec check(cartid);
- The given cart id is not existed so we will get a message with " cart id in invalid"

Case 2:

- Input: Cartid=9;
- Exec check(Cartid,deliverymethod,checkouttime)
- At check out time if there's discount then add that discount at the discount rate and check delivery method whether it is pickup or delivery here the cart id with 9 is pickup so no need to add extra price for the given restaurant location the tax is 1.2
  - Price=20*4 (price*quantity) So sum is 80
  - Discount rate = price*0.9
  - Tax is 1.2 so total is 0.9*80*1.2=86.4
    Customer has to pay 86.4

Feature 12:

Input: cart ID, order time, deliver method (1 deliver, 2 pickup), an estimated time to deliver or pickup, tip, and a payment method
Output: First we will validate the Cart id it if it is valid then we will call the procedure in feature 11 and add the tip to it before making the payments table populate the input parameters into shopping cart and payments table if the quantity is reduced update the cart dish table with order number of quantities in addition to that populate the customer's message into the message table.

Case 1:

- Input Cart id =11;
- Exec check(cartid);
- The given cart id is not existed so we will get a message with " cart id in invalid"

Case 2:

- Input: Cartid=9;
- Exec check(Cartid), exec call procedure(11)
- Exec fetch(orderid,customerid,cart's customer id,restaurantid,ordertime)
  - Exec fetch(Estimated time,status in progress,delivery method,delivery fee,tax, total) from call procedure (11) is **86.4**
  - Exec fetch input(tip) is **3.6** now total is 90
  - Exec insert(dishes) into shopping cart and exec delete(shopping cart,dishes)
  - Exec insert(payment record) into payments_table
  - With payment id =816, customerid= 5 Payment time=06-OCT-22 12.05.00.000000000 PM, order ID: 579, Total=90 Payment Method: debit

<u>Feature 13</u>

Input : CustomerID , category names , review score and wait time

Output : The procedure first checks whether the given customer id is valid or not. If invalid we print invalid customer .If id exists the procedure return all the restaurants which satisfy the conditions a) under one of the input categories;b) average review score greater or equal to the minimal score,c) a wait time less or equal to the input wait time.d) having a zip code either the same as the customer's zip code or differ only by the last digit . Last step is to print out the name of the restaurant, address, status, average review score, zip code, and average wait time.

Case 1 :

- **Input :** let us consider customer id is 6

- List of category names be(pizza , seafood , Mexican)
- Minimal review score : 1
- Wait time 12
- Procedure AdvancedSearch(customer ID, a list of category names, a minimal review score, and a wait time.)
- As custId ‘6’ is not available in our database

Output : Invalid Customer

Case 2 :

Input : let us consider customer id is 1

- List of category names be(pizza , seafood , Mexican)
- Minimal review score : 1
- Wait time 12
- Procedure AdvancedSearch(customer ID, a list of category names, a minimal review score, and a wait time.)
- Customer ID ‘1’ exists .
- Procedure returns restaurant name which satisfy the above conditions

Output : dbms_output.put_line(name of restaurant, address, status, average review score, zip code, and average wait time.);

- Persis 2873Eldon Open 1 21229 10.2 1

Feature 14 :

Input : Customer ID

Output : First the procedure checks whether the customer id is valid or not . if not , it prints out a message as 'Invalid customer' and then stops. If custid exists the procedure finds out the restaurants that customer has placed an order and print out the restaurant ids. In the third step , the procedure finds customers who have placed orders in any restaurant in step 2 and prints out these customers IDs . In step 4 , it finds other restaurants these customers (in step 3) go to and print out id and names of these restaurants, their addresses and average reviews.

Case 1 :

- **Input :** 6
- Procedure Restaurant_recommendation(CustID).
- As Cust ID with ‘6’ does not exist.
- Output : Invalid Customer.

Case 2:

- **Input :** 1
- Procedure Restaurant_recommendation(CustID).
- Customer exists and procedure finds the restaurant that customer has placed an order and prints the restaurant id .
- Output : dbms_output.put_line(restaurantid);
- 101
- Next is to find customers who have placed an order in a restaurant which has restaurant id 101 and print out those customer ids.
- Dbms_output.put_line (customerid);
- Finding other restaurants these customers go to and Print out id and names of these restaurants, their addresses and average reviews
- Dbms_output.put_line(restaurantid , restaurantname , address , averagereviews).
- 101 Persis 2873 Eldon 1