drop table MESSAGE\_TABLE cascade constraints;

drop table PAYMENT\_DETAILS cascade constraints;

drop table DISHES\_IN\_ORDER cascade constraints;

drop table CUSTOMER\_ORDER cascade constraints;

drop table DISHES\_IN\_CART cascade constraints;

drop table CART cascade constraints;

drop table CUSTOMER\_REVIEW cascade constraints;

drop table DISHES cascade constraints;

drop table RESTAURANT\_CATEGORY cascade constraints;

drop table RESTAURANT\_LIST cascade constraints;

drop table FOOD\_CATEGORY cascade constraints;

drop table CUSTOMER\_DISCOUNT cascade constraints;

drop table SALES\_TAX cascade constraints;

drop table DISCOUNT\_INFO cascade constraints;

drop table CUSTOMER\_INFO cascade constraints;

-------------------------------------------------------------------------

create table CUSTOMER\_INFO (

customer\_id int primary key,

customer\_name varchar2(100),

customer\_address varchar2(100),

customer\_zipcode number,

customer\_state varchar2(50),

customer\_email varchar2(100),

customer\_credit varchar(50)

);

insert into CUSTOMER\_INFO

values (1, 'Stacey', '13b Leafcrest Ln, Fairfax, Virginia', 22033, 'Virginia', 'stacey@gmail.com', 102);

insert into CUSTOMER\_INFO

values (2, 'Namrata', '5346 Jamestowne Ct, Baltimore, Maryland', 21229, 'Maryland', 'namrata@gmail.com',108);

insert into CUSTOMER\_INFO

values (3, 'Sheryl', '102 Des Plaines, Chicago, Illinois', 60018, 'Illinois', 'sheryl@gmail.com', 103);

insert into CUSTOMER\_INFO

values (4, 'Joanna', '502 Scount Street, Tampa, Florida', 33592, 'Florida', 'joanna@gmail.com', 105);

insert into CUSTOMER\_INFO

values (5, 'Sharon', '9030 Helena Ct, Milwaukee, Wisconsin', 53224, 'Wisconsin', 'sharon@gmail.com', 106);

insert into CUSTOMER\_INFO

values (6, 'Sudheer', '14 Horadan Way, Boston, Massachusetts', 02111, 'Massachusetts', 'susheer@gmail.com', 109);

insert into CUSTOMER\_INFO

values (7, 'Nora', '1343 Oak Cir, Arnold, California', 95223, 'California', 'nora@gmail.com', 97);

insert into CUSTOMER\_INFO

values (8, 'Kitoli', '205 Varco Drive, Austin, Texas', 78738, 'Texas', 'kitoli@gmail.com', 96);

insert into CUSTOMER\_INFO

values (9, 'Prabhakar', '46w Princeton Ave, Denver, Colorado', 80014, 'Colorado', 'prabhkar@gmail.com', 83);

insert into CUSTOMER\_INFO

values (10, 'Amrita', '23 Golder St, Albany, New York', 12208, 'New York', 'amrita@gmail.com', 72);

--------------------------------------------------------------------------

create table DISCOUNT\_INFO (

discount\_id int primary key,

discount\_description varchar2(500),

discount\_type number,

discount\_amount number

);

insert into DISCOUNT\_INFO values (11, 'free delivery', 1, 54);

insert into DISCOUNT\_INFO values (12, 'fixed percent off the total charge', 2, 100);

insert into DISCOUNT\_INFO values (13, 'fixed amount off the total charge', 3, 20);

insert into DISCOUNT\_INFO values (14, 'fixed percent off the total charge', 2, 105);

insert into DISCOUNT\_INFO values (15, 'free delivery', 1, 60);

insert into DISCOUNT\_INFO values (16, 'fixed amount off the total charge', 3, 30);

insert into DISCOUNT\_INFO values (17, 'free delivery', 1, 72);

insert into DISCOUNT\_INFO values (18, 'free delivery', 1, 85);

insert into DISCOUNT\_INFO values (19, 'fixed amount off the total charge', 3, 80);

insert into DISCOUNT\_INFO values (20, 'free delivery', 1, 90);

---------------------------------------------------------------------------

create table SALES\_TAX (

state varchar2(50) primary key,

tax\_rate number

);

insert into SALES\_TAX values ('Maryland',2);

insert into SALES\_TAX values ('Massachusetts',5);

insert into SALES\_TAX values ('Virginia',3);

insert into SALES\_TAX values ('Illinois',6);

insert into SALES\_TAX values ('Florida',5);

insert into SALES\_TAX values ('Colorado',4);

insert into SALES\_TAX values ('Texas',6);

insert into SALES\_TAX values ('California',7);

insert into SALES\_TAX values ('New York',5);

insert into SALES\_TAX values ('Wisconsin',5);

----------------------------------------------------------------------------

create table CUSTOMER\_DISCOUNT (

customer\_id int,

discount\_id int,

discount\_startdate timestamp,

discount\_enddate timestamp,

foreign key (customer\_id) REFERENCES CUSTOMER\_INFO(customer\_id),

foreign key (discount\_id) REFERENCES DISCOUNT\_INFO(discount\_id)

);

insert into CUSTOMER\_DISCOUNT values (1, 11, '1-Jan-22', '1-Feb-22');

insert into CUSTOMER\_DISCOUNT values (2, 12, '1-Feb-22', '1-Mar-22');

insert into CUSTOMER\_DISCOUNT values (3, 13, '1-Mar-22', '1-Apr-22');

insert into CUSTOMER\_DISCOUNT values (4, 14, '1-Apr-22', '1-May-22');

insert into CUSTOMER\_DISCOUNT values (5, 15, '1-May-22', '1-Jun-22');

insert into CUSTOMER\_DISCOUNT values (6, 16, '1-Jun-22', '1-Jul-22');

insert into CUSTOMER\_DISCOUNT values (7, 17, '1-Jul-22', '1-Aug-22');

insert into CUSTOMER\_DISCOUNT values (8, 18, '1-Aug-22', '1-Sep-22');

insert into CUSTOMER\_DISCOUNT values (9, 19, '1-Sep-22', '1-Oct-22');

insert into CUSTOMER\_DISCOUNT values (10, 20, '1-Oct-22', '1-Nov-22');

------------------------------------------------------------------------------

create table FOOD\_CATEGORY (

category\_id int primary key,

category\_name varchar2(100)

);

insert into FOOD\_CATEGORY values (21, 'Bakery');

insert into FOOD\_CATEGORY values (22, 'Indian');

insert into FOOD\_CATEGORY values (23, 'Mexican');

insert into FOOD\_CATEGORY values (24, 'Japanese');

insert into FOOD\_CATEGORY values (25, 'Arabian');

insert into FOOD\_CATEGORY values (26, 'Jamaican');

insert into FOOD\_CATEGORY values (27, 'Italian');

insert into FOOD\_CATEGORY values (28, 'Ice cream');

insert into FOOD\_CATEGORY values (29, 'Vegan');

insert into FOOD\_CATEGORY values (30, 'Breakfast');

-------------------------------------------------------------------------------

create table RESTAURANT\_LIST (

restaurant\_id int primary key,

restaurant\_name varchar2(200),

restaurant\_address varchar2(50),

restaurant\_phno number,

restaurant\_status varchar2(10), --open or closed

restaurant\_zipcode number,

state varchar2(50),

average\_waittime timestamp,

average\_reviewscore number,

foreign key (state) REFERENCES SALES\_TAX(state)

);

insert into RESTAURANT\_LIST

values (31, 'Jamaican Curries','Catonsville, Baltimore, Maryland',4109405043,'open',21229,'Maryland',to\_timestamp('30','MI'),3);

insert into RESTAURANT\_LIST

values (32, 'Dunkin Donuts','Pink Ave, Boston, Massachusetts',9885775899,'open',02111,'Massachusetts',to\_timestamp('3','MI'),4.5);

insert into RESTAURANT\_LIST

values (33, 'Taco bell','Mallow hills, Tampa, Florida',9490993589,'close',33592,'Florida',to\_timestamp('4','MI'),4.5);

insert into RESTAURANT\_LIST

values (34, 'Cake Lmore','Williston, Chicago, Illinois',9949996405,'open',60018,'Illinois',to\_timestamp('20','MI'),4);

insert into RESTAURANT\_LIST

values (35, 'Lets Vegan','Chatilly Blvd, Fairfax, Virginia',7075803589,'open',22033,'Virginia',to\_timestamp('45','MI'),2.5);

insert into RESTAURANT\_LIST

values (36, 'Coffee cup','Wellington, Milwaukee, Wisconsin',2722256489,'open',53224,'Wisconsin',to\_timestamp('15','MI'),3.5);

insert into RESTAURANT\_LIST

values (37, 'Paradise Biryani','Lakshmi St, Austin, Texas',9848421537,'close',78738,'Texas',to\_timestamp('10','MI'),5);

insert into RESTAURANT\_LIST

values (38, 'Cream Stone','South Wickham, Arnold, California',8700152392,'close',95223,'California',to\_timestamp('6','MI'),4.5);

insert into RESTAURANT\_LIST

values (39, 'Gismat Mandi','Square St, Denver, Colorado',6678415623,'open',80014,'Colorado',to\_timestamp('50','MI'),5);

insert into RESTAURANT\_LIST

values (40, 'Chuby chu','Maiden Choice, Albany, New York',5408409636,'open',12208,'New York',to\_timestamp('25','MI'),3);

---------------------------------------------------------------------------------

create table RESTAURANT\_CATEGORY (

restaurant\_id int,

category\_id int,

category\_name varchar2(100),

foreign key (restaurant\_id) REFERENCES RESTAURANT\_LIST(restaurant\_id),

foreign key (category\_id) REFERENCES FOOD\_CATEGORY(category\_id)

);

insert into RESTAURANT\_CATEGORY values (31, 26, 'Jamaican');

insert into RESTAURANT\_CATEGORY values (32, 30, 'Breakfast');

insert into RESTAURANT\_CATEGORY values (33, 23, 'Mexican');

insert into RESTAURANT\_CATEGORY values (34, 21, 'Bakery');

insert into RESTAURANT\_CATEGORY values (35, 29, 'Vegan');

insert into RESTAURANT\_CATEGORY values (36, 27, 'Italian');

insert into RESTAURANT\_CATEGORY values (37, 22, 'Indian');

insert into RESTAURANT\_CATEGORY values (38, 28, 'Ice cream');

insert into RESTAURANT\_CATEGORY values (39, 25, 'Arabian');

insert into RESTAURANT\_CATEGORY values (40, 24, 'Japanese');

----------------------------------------------------------------------------------

create table DISHES (

dish\_id int primary key,

restaurant\_id int,

dish\_name varchar2(100),

dish\_price int,

foreign key (restaurant\_id) REFERENCES RESTAURANT\_LIST(restaurant\_id)

);

insert into DISHES values (41,31,'Curry shrimp, stew peas and rice',16);

insert into DISHES values (42,32,'Eggs, bacon and pancakes',8);

insert into DISHES values (43,33,'Hard shell beef tacos',5);

insert into DISHES values (44,34,'Chocolate truffle bday cake',30);

insert into DISHES values (45,35,'Vegan hamburger',15);

insert into DISHES values (46,36,'Red sauce pasta',25);

insert into DISHES values (47,37,'Mutton dum biryani',24);

insert into DISHES values (48,38,'Death by chocolate',18);

insert into DISHES values (49,39,'Gravy chicken mandi',40);

insert into DISHES values (50,40,'Chicken Ramen',10);

-----------------------------------------------------------------------------------

create table CUSTOMER\_REVIEW (

review\_id int primary key,

customer\_id int,

restaurant\_id int,

review\_date timestamp,

review\_score int,

review\_comments varchar2(500),

foreign key (customer\_id) REFERENCES CUSTOMER\_INFO(customer\_id),

foreign key (restaurant\_id) REFERENCES RESTAURANT\_LIST(restaurant\_id)

);

insert into CUSTOMER\_REVIEW values (51, 1, 31,'15-Jan-22',3,'Happy with the quantity, would appreciate more flavour');

insert into CUSTOMER\_REVIEW values (52, 2, 32,'15-Feb-22',5,'Good breakfast with reasonable price');

insert into CUSTOMER\_REVIEW values (53, 3, 33,'15-Mar-22',4,'Taste is good but not authentic');

insert into CUSTOMER\_REVIEW values (54, 4, 34,'15-Apr-22',4.5,'Chocolate cake could never go wrong, would love to try other items from this bakery');

insert into CUSTOMER\_REVIEW values (55, 5, 35,'15-May-22',2,'Not very happy');

insert into CUSTOMER\_REVIEW values (56, 6, 36,'15-Jun-22',3.5,'Good');

insert into CUSTOMER\_REVIEW values (57, 7, 37,'15-Jul-22',5,'Out of the world, will definitely order once again');

insert into CUSTOMER\_REVIEW values (58, 8, 38,'15-Aug-22',5,'Amazing');

insert into CUSTOMER\_REVIEW values (59, 9, 39,'15-Sep-22',5,'Satisfied with quantity and taste');

insert into CUSTOMER\_REVIEW values (60, 10, 40,'15-Oct-22',4,'Loved it');

------------------------------------------------------------------------------------

create table CART (

cart\_id int primary key,

customer\_id int,

restaurant\_id int,

foreign key (customer\_id) REFERENCES CUSTOMER\_INFO(customer\_id),

foreign key (restaurant\_id) REFERENCES RESTAURANT\_LIST(restaurant\_id)

);

---1,3,4,5,6 (DELIVERY)

---2,7,8,9,10 (PICKUP)

insert into CART values (61, 1, 31);

insert into CART values (62, 3, 33);

insert into CART values (63, 4, 34);

insert into CART values (64, 5, 35);

insert into CART values (65, 6, 36);

insert into CART values (66, 2, 32);

insert into CART values (67, 7, 37);

insert into CART values (68, 8, 38);

insert into CART values (69, 9, 39);

insert into CART values (70, 10, 40);

-------------------------------------------------------------------------------------

create table DISHES\_IN\_CART (

restaurant\_id int,

dish\_id int,

quantity number,

foreign key (restaurant\_id) REFERENCES RESTAURANT\_LIST(restaurant\_id),

foreign key (dish\_id) REFERENCES DISHES(dish\_id)

);

insert into DISHES\_IN\_CART values (31, 41, 2);

insert into DISHES\_IN\_CART values (33, 43, 20);

insert into DISHES\_IN\_CART values (34, 44, 1);

insert into DISHES\_IN\_CART values (35, 45, 3);

insert into DISHES\_IN\_CART values (36, 46, 2);

insert into DISHES\_IN\_CART values (32, 42, 1);

insert into DISHES\_IN\_CART values (37, 47, 2);

insert into DISHES\_IN\_CART values (38, 48, 4);

insert into DISHES\_IN\_CART values (39, 49, 2);

insert into DISHES\_IN\_CART values (40, 50, 1);

--------------------------------------------------------------------------------------

create table CUSTOMER\_ORDER (

order\_id int primary key,

customer\_id int,

restaurant\_id int,

order\_time timestamp,

delivery\_time timestamp,

est\_time timestamp,

delivery\_status varchar2(50),

payment\_status varchar2(100),

order\_totalcost int,

delivery\_method int,

delivery\_totalcost int,

pickup\_totalcost int,

foreign key (customer\_id) REFERENCES CUSTOMER\_INFO(customer\_id),

foreign key (restaurant\_id) REFERENCES RESTAURANT\_LIST(restaurant\_id)

);

---1 is delivery, 2 as pickup

---in progress, delivered or canceled

---paid or not paid

---total cost = prices for all ordered dishes + a delivery fee + tip + sales tax (based on the state)

---total cost = prices of dishes + sales tax

insert into CUSTOMER\_ORDER

values (71, 1,31,to\_timestamp('10-Jan-22 13:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-Jan-22 13:25','DD-Mon-RR HH24:MI'),to\_timestamp('10-Jan-22 13:30','DD-Mon-RR HH24:MI'),'delivered','paid',32,1,37.74,NULL);

insert into CUSTOMER\_ORDER

values (72, 3,33,to\_timestamp('10-Mar-22 18:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-Mar-22 18:13','DD-Mon-RR HH24:MI'),to\_timestamp('10-Mar-22 18:20','DD-Mon-RR HH24:MI'),'in progress','paid',100,1,111.3,NULL);

insert into CUSTOMER\_ORDER

values (73, 4,34,to\_timestamp('10-Apr-22 17:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-Apr-22 13:05','DD-Mon-RR HH24:MI'),to\_timestamp('10-Apr-22 13:10','DD-Mon-RR HH24:MI'),'cancelled','not paid',30,1,36.75,NULL);

insert into CUSTOMER\_ORDER

values (74, 5,35,to\_timestamp('10-May-22 12:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-May-22 13:15','DD-Mon-RR HH24:MI'),to\_timestamp('10-May-22 13:20','DD-Mon-RR HH24:MI'),'delivered','paid',75,1,78.75,NULL);

insert into CUSTOMER\_ORDER

values (75, 6,36,to\_timestamp('10-Jun-22 11:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-Jun-22 11:30','DD-Mon-RR HH24:MI'),to\_timestamp('10-Jun-22 11:40','DD-Mon-RR HH24:MI'),'in progress','paid',50,1,52.5,NULL);

insert into CUSTOMER\_ORDER

values (76, 2,32,to\_timestamp('10-Jul-22 07:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-Jul-22 07:10','DD-Mon-RR HH24:MI'),to\_timestamp('10-Jul-22 07:15','DD-Mon-RR HH24:MI'),'in progress','paid',8,2,NULL,8.16);

insert into CUSTOMER\_ORDER

values (77, 7,37,to\_timestamp('10-Aug-22 21:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-Aug-22 21:30','DD-Mon-RR HH24:MI'),to\_timestamp('10-Aug-22 21:25','DD-Mon-RR HH24:MI'),'in progress','paid',48,2,NULL,51.36);

insert into CUSTOMER\_ORDER

values (78, 8,38,to\_timestamp('10-Sep-22 16:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-Sep-22 16:10','DD-Mon-RR HH24:MI'),to\_timestamp('10-Sep-22 16:15','DD-Mon-RR HH24:MI'),'cancelled','not paid',72,2,NULL,76.32);

insert into CUSTOMER\_ORDER

values (79, 9,39,to\_timestamp('10-Oct-22 12:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-Oct-22 12:05','DD-Mon-RR HH24:MI'),to\_timestamp('10-Oct-22 12:10','DD-Mon-RR HH24:MI'),'in progress','paid',80,2,NULL,83.2);

insert into CUSTOMER\_ORDER

values (80, 10,40,to\_timestamp('10-Nov-22 19:00','DD-Mon-RR HH24:MI'),to\_timestamp('10-Nov-22 19:10','DD-Mon-RR HH24:MI'),to\_timestamp('10-Nov-22 19:20','DD-Mon-RR HH24:MI'),'in progress','paid',10,2,NULL,10.5);

----------------------------------------------------------------------------------

create table DISHES\_IN\_ORDER (

order\_id int,

dish\_id int,

foreign key (order\_id) REFERENCES CUSTOMER\_ORDER(order\_id),

foreign key (dish\_id) REFERENCES DISHES(dish\_id)

);

insert into DISHES\_IN\_ORDER values (71, 41);

insert into DISHES\_IN\_ORDER values (72, 43);

insert into DISHES\_IN\_ORDER values (73, 44);

insert into DISHES\_IN\_ORDER values (74, 45);

insert into DISHES\_IN\_ORDER values (75, 46);

insert into DISHES\_IN\_ORDER values (76, 42);

insert into DISHES\_IN\_ORDER values (77, 47);

insert into DISHES\_IN\_ORDER values (78, 48);

insert into DISHES\_IN\_ORDER values (79, 49);

insert into DISHES\_IN\_ORDER values (80, 50);

-----------------------------------------------------------------------------------

create table PAYMENT\_DETAILS (

payment\_id int primary key,

customer\_id int,

payment\_time timestamp,

order\_id int,

payment\_amount int,

payment\_method varchar2(50),

foreign key (customer\_id) REFERENCES CUSTOMER\_INFO(customer\_id),

foreign key (order\_id) REFERENCES CUSTOMER\_ORDER(order\_id)

);

insert into PAYMENT\_DETAILS values (81, 1,to\_timestamp('10-Jan-22 13:00','DD-Mon-RR HH24:MI'),71,37.74,'credit card');

insert into PAYMENT\_DETAILS values (82, 3,to\_timestamp('10-Mar-22 18:00','DD-Mon-RR HH24:MI'),72,111.3,'credit card');

insert into PAYMENT\_DETAILS values (83, 4,to\_timestamp('10-Apr-22 17:00','DD-Mon-RR HH24:MI'),73,36.75,NULL);

insert into PAYMENT\_DETAILS values (84, 5,to\_timestamp('10-May-22 12:00','DD-Mon-RR HH24:MI'),74,78.75,'paypal');

insert into PAYMENT\_DETAILS values (85, 6,to\_timestamp('10-Jun-22 11:00','DD-Mon-RR HH24:MI'),75,52.5,'debit card');

insert into PAYMENT\_DETAILS values (86, 2,to\_timestamp('10-Jul-22 07:00','DD-Mon-RR HH24:MI'),76,8.16,'applepay');

insert into PAYMENT\_DETAILS values (87, 7,to\_timestamp('10-Aug-22 21:00','DD-Mon-RR HH24:MI'),77,51.36,'paypal');

insert into PAYMENT\_DETAILS values (88, 8,to\_timestamp('10-Sep-22 16:00','DD-Mon-RR HH24:MI'),78,76.32,NULL);

insert into PAYMENT\_DETAILS values (89, 9,to\_timestamp('10-Oct-22 12:00','DD-Mon-RR HH24:MI'),79,83.2,'debit card');

insert into PAYMENT\_DETAILS values (90, 10,to\_timestamp('10-Nov-22 19:00','DD-Mon-RR HH24:MI'),80,10.5,'applepay');

------------------------------------------------------------------------------------

create table MESSAGE\_TABLE (

message\_id int primary key,

customer\_id int,

message\_time timestamp,

message\_body varchar2(1000),

foreign key (customer\_id) REFERENCES CUSTOMER\_INFO(customer\_id)

);

insert into MESSAGE\_TABLE values (91,1,to\_timestamp('10-Jan-22 13:00','DD-Mon-RR HH24:MI'),'make it spicy');

insert into MESSAGE\_TABLE values (92,2,to\_timestamp('10-Jul-22 07:00','DD-Mon-RR HH24:MI'),'hot');

insert into MESSAGE\_TABLE values (93,3,to\_timestamp('10-Mar-22 18:00','DD-Mon-RR HH24:MI'),'extra sauce');

insert into MESSAGE\_TABLE values (94,4,to\_timestamp('10-Apr-22 17:00','DD-Mon-RR HH24:MI'),NULL);

insert into MESSAGE\_TABLE values (95,5,to\_timestamp('10-May-22 12:00','DD-Mon-RR HH24:MI'),'salad on the side');

insert into MESSAGE\_TABLE values (96,6,to\_timestamp('10-Jun-22 11:00','DD-Mon-RR HH24:MI'),'no bread');

insert into MESSAGE\_TABLE values (97,7,to\_timestamp('10-Aug-22 21:00','DD-Mon-RR HH24:MI'),'medium spice');

insert into MESSAGE\_TABLE values (98,8,to\_timestamp('10-Sep-22 16:00','DD-Mon-RR HH24:MI'),NULL);

insert into MESSAGE\_TABLE values (99,9,to\_timestamp('10-Oct-22 12:00','DD-Mon-RR HH24:MI'),'no salad');

insert into MESSAGE\_TABLE values (100,10,to\_timestamp('10-Nov-22 19:00','DD-Mon-RR HH24:MI'),'medium spice');

------------------------------------------------------------------------------------

select \* from CUSTOMER\_INFO;

select \* from DISCOUNT\_INFO;

select \* from SALES\_TAX;

select \* from CUSTOMER\_DISCOUNT;

select \* from FOOD\_CATEGORY;

select \* from RESTAURANT\_LIST;

select \* from RESTAURANT\_CATEGORY;

select \* from DISHES;

select \* from CUSTOMER\_REVIEW;

select \* from CART;

select \* from DISHES\_IN\_CART;

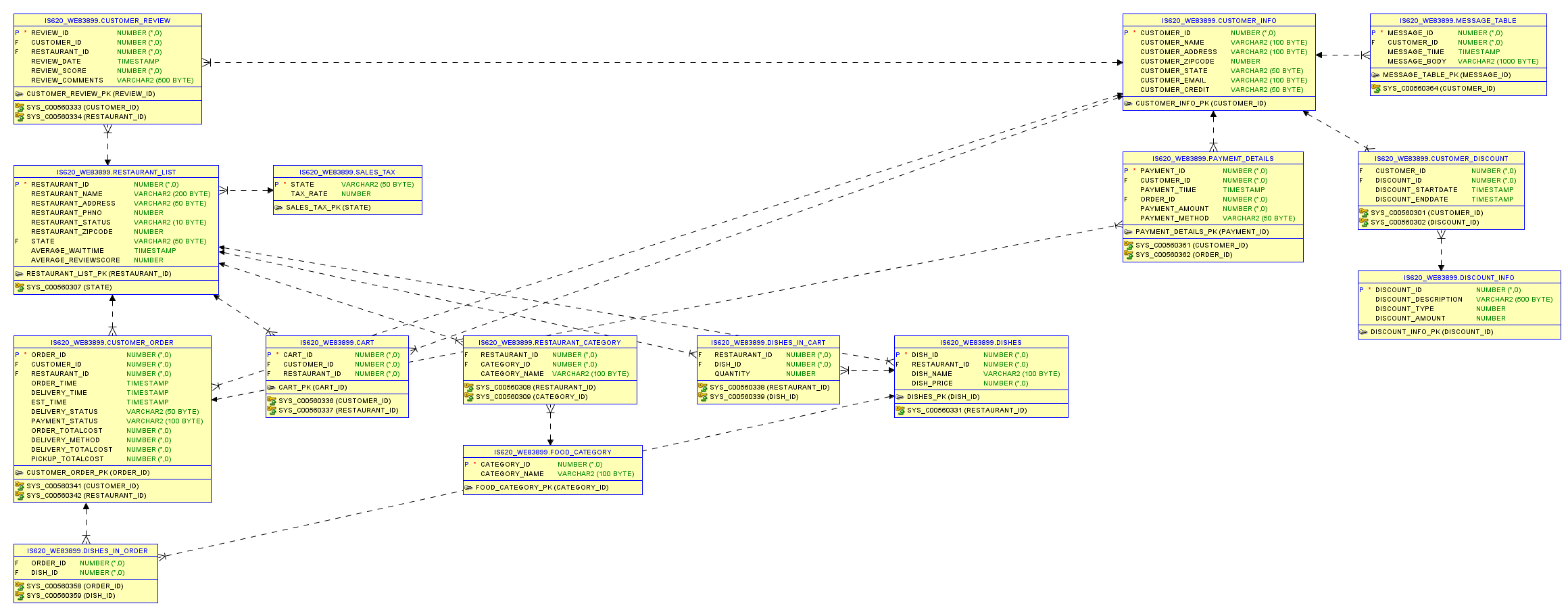
select \* from CUSTOMER\_ORDER;

select \* from DISHES\_IN\_ORDER;

select \* from PAYMENT\_DETAILS;

select \* from MESSAGE\_TABLE;

**ER DIAGRAM:**



****

**FEATURE SPECIFICATIONS:**

**FEATURE 1:**

create a new customer

Input: customer name, address, state, zip, email.

Output: screen output of 'the client already exists' if a client with the same email exists and update of address, zip code and state. Otherwise, a new row inserted client table and a screen output of the newly assigned client ID.

**Example of calling this feature:**

---- regular case

exec ADD\_CUSTOMER ('Sahithi', '13459 Mountridge ave, Westland, Maryland', 'Maryland','21042', 'sahithi@gmail.com');

-- existing case (adding the same client again)

select \* from customer;

exec ADD\_CUSTOMER ('Joshua', '13459 Mountridge ave, Westland, Maryland', 'Maryland','21042', 'joshua@gmail.com');

**FEATURE 2:**

To check whether there is a customer with the given email id.

Input: email id

Output: screen output ‘no such customer’ if there is no customer with that email id. Else, print out the profile of the customer, including 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.

**Example of calling this feature**: exec CHECK\_EMAIL(‘stacey@gmail.com’); ----regular case

**FEATURE 3:**

Search restaurant by category.

Input: category\_name or part of category\_name

Output: name of restaurant, average review score, average wait time, and zip code for restaurants that are open if it matches the input.

**Example of calling this feature:** exec REST\_CATEGORY(‘fastfood’); ----regular case

**FEATURE 4:**

Show dishes offered by a restaurant

Input: restaurant ID

Output: All dishes in this restaurant, along with dish name and price. If no such restaurant id exits, then print ‘No such restaurant’.

**Example of calling this feature**: exec DISHES\_OFFERED (31);

**FEATURE 5:**

Show all dishes in shopping cart

Input: cart ID

Output: print out every dish in the shopping cart, including dish name, price, quantity. If cart\_id is invalid, then print ‘invalid cart id’.

**Example of calling this feature**: exec CART\_DISHES (64);

**FEATURE 6:**

Remove a dish from shopping cart

Input: dish ID and cart ID.

Output: First check whether the cart with the given ID has that dish. If not print a message 'Invalid input'. 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 and print a message saying ‘quantity reduced’. If the quantity is one, delete that row from the cart and print out 'dish removed'.

**Example of calling this feature**: exec REMOVE\_DISH (45, 65);

**FEATURE 7:**

Update status of an order.

Input: order ID, new status (1 is in progress, 2 is delivered, 3 is cancelled), and input time.

Output:

1) First checks whether the order ID is valid. If not print a message saying invalid order id.

2) Update the status of the order to the input status. In case new status is in progress, no additional action is needed. ----Procedure ends here.

3) In case new status is ‘delivered’, insert a message into message table for the corresponding customer, with message time as input time, and message body saying 'Your order X has been delivered!' where X is the order ID.

4) In case new status is ‘cancelled’, update the status to cancelled, insert a message into message table for the corresponding customer, with message time as input time, and message body saying 'Your order X has been cancelled and refund issued!' where X is the order ID. Also insert into payment table a new refund record with new payment id and reduce the amount for the order from the total amount.

**Example of calling this feature:**

In Progress: exec UPDATE\_ORDER\_STATUS (71, 1, systimestamp);

Delivered: exec UPDATE\_ORDER\_STATUS (72, 2, systimestamp);

Cancelled: exec UPDATE\_ORDER\_STATUS (73, 3, systimestamp);

**FEATURE 8:**

To enter a review in customer\_review table.

Input: customer ID, a restaurant ID, a review date, a review score and review comment

Output:

1) first checks if the customer ID is valid. If not print a message saying invalid customer ID.

2) Check if the restaurant ID is valid. If not print a message saying invalid restaurant ID.

3) if both are valid, insert a row into review table with the input customer id, restaurant ID, review date, score and comment.

4) update the average review score of the restaurant to reflect the new review.

**Example of calling this feature:**

exec REVIEW (1, 31, sysdate, 3,’ Happy with the quantity, would appreciate more flavour’);

**FEATURE 9:**

Display all reviews of a restaurant

Input: restaurant ID

Output: First checks whether the restaurant ID is valid. If not print a message ‘invalid restaurant id’. Then print out all reviews of the restaurant, including review date, score, and comment.

**Example of calling this feature**: exec DISPLAY\_REVIEWS (32);

**FEATURE 10:**

Add a dish to shopping cart

Input: customer ID, restaurant ID, and a dish ID.

Output:

1. First check whether the customer ID is valid. If not print out a message no such customer.
2. Then check whether the restaurant ID is valid and the restaurant is open. If not print out invalid restaurant ID or the restaurant is closed.
3. Finally check the dish whether it belongs to the input restaurant. If it does not print out message invalid dish ID.
4. Otherwise, where there is an existing shopping cart for the customer. If the cart does not exit, create a new cart for the customer and restaurant and print out the new cart ID.
5. Now you can check whether the dish is already in the cart. If so, just increase the quantity by one. Otherwise insert a new row to the table keeps dishes in a cart.

**Example of calling this feature**: exec ADD\_DISHTOCART (1, 31, 41);

**FEATURE 11:**

Compute total amount for dishes in a cart.

Input: cart ID and a checkout time, delivery method (1 deliver, 2 pickup).

Output: checks whether the cart ID is valid. If not, it prints a message invalid cart ID. If cart is valid, print the total amount.

The total amount includes the price for each dish in the cart and quantity. If discount is applied, reduce discount amount to the total amount. Delivery fee if the order is not pickup, tip and state tax.

**Example of calling this feature:**

Delivery: exec TOTAL\_AMOUNT\_FOR DISHES (61, systimestamp, 1);

Pickup: exec TOTAL\_AMOUNT\_FOR DISHES (62, systimestamp, 2);

**FEATURE 12:**

Generate an order with dishes in a shopping cart.

Input: The input is a cart ID, order time, deliver method (1 deliver, 2 pickup), an estimated time to deliver or pickup, tip, and a payment method (1 credit/debit, 2 apple pay, 3 paypal).

Output:

1) check if the cart ID is valid, if not print a message invalid cart ID and stop.

2) insert a row into orders table with a newly generated order ID, customer ID as the cart's customer ID, restaurant ID as the cart's restaurant ID, order time as the input order time, delivery time is null, estimated time as the input estimated time, status is in progress, delivery method is the input delivery method, delivery fee, tax, and total are returned in feature 11, and tip is the input tip.

4) insert dishes in the shopping cart into the table that stores dishes in the order. Delete the shopping cart row and dishes in the cart.

5) insert into message table a message with customer ID as the customer ID associated with the cart, message time as the order time, and body as

'A new order X is placed at Restaurant Y with estimated time of Z and amount A', where X is the order ID, Y is the name of the restaurant, Z as input estimated time (in minutes), and A is total amount.

6) insert a payment record to payment table with order ID as the order ID, payment time as order time, payment amount as total computed in step 2, and payment method as input payment method.

**Example of calling this feature:**

Credit/Debit: exec GENERATE\_ORDER (61, systimestamp, 1, systimestamp, 2, 1);

Apple pay: exec GENERATE\_ORDER (61, systimestamp, 1, systimestamp, 2, 2);

Paypal: exec GENERATE\_ORDER (61, systimestamp, 1, systimestamp, 2, 3);

**FEATURE 13:**

Printing out the name of the restaurant for a given input (Advanced search)

Input: customer ID, a list of category names, a minimal review score, and a wait time, zipcode

Output: If customer ID is not valid, print invalid customer

Else, the procedure returns all restaurants that satisfy ALL of the following conditions:

1. under one of the input categories;

b) with an 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

**Example of calling this feature:**

exec ADVANCED\_SEARCH (1, varray\_catname, 2, to\_timestamp('30','MI'), substr(21229, 1,4));

Where varray\_catname is a vary which will be declared in the procedure.

**FEATURE 14:**

Restaurant recommendation.

Input: customer ID

Output:

1) check whether the customer ID is valid. If not print an error message and stop.

2) Find restaurants that customer has placed an order. Exclude the same customer in the input. Print out these restaurants' IDs.

3) Find customers who have placed orders in any restaurant in step 2). Print out these customers IDs.

4) Find other restaurants these customers (in step 3) go to. Exclude those restaurants in step 2) (i.e., the customer already visited).

5) Print out id and names of these restaurants, their addresses and average reviews.

**Example of calling this feature:** Exec REST\_RECOM (2);

Note: We will have to insert more data to check whether all conditions in the features are satisfied.