

# Cloud Kitchen Management System

## **ABSTRACT:**

The project is based on the concept of cloud kitchen as its quicker and cheaper to launch a business if the customer interaction is virtual. In this way, both the seller and buyer gets high perks of benefits cutting down on equipment cost, deal with less legal paperwork, and can start selling almost immediately. The database consists of six tables that includes, customers, kitchen, dishes, orders, reviews, deliverer. The consumer gets information about the order, kitchen, reviews and dishes. The admins get the customer information, orders made by them, and feedbacks.

## **BUSINESS PROBLEM:**

In recent times, the concept of a cloud kitchen has recently gained momentum, owing to a fact in diner behavior caused by COVID-19 pandemic, in which people prefer food delivery over dining out. A cloud kitchen is an ideal solution for people who want to open their own restaurant but are hampered by a lack of resources. Cloud kitchens, a business model that promises to optimize and expand delivery services at a low cost, have resulted in the emergence of the food delivery revolution. Hence, bringing restaurants to a new level where seamless delivery is prioritized, costs are reduced, and profits are maximized, we have created a database application to bring home kitchen and customer together.

## **PROBLEM STATEMENT:**

The absence of an organized and efficient system for online food delivery service hinders seamless operations, effective order management, and customer satisfaction. There is a pressing need to create a comprehensive database and platform that can streamline the entire process, from order placement to delivery, while ensuring data accuracy and providing an optimal user experience.

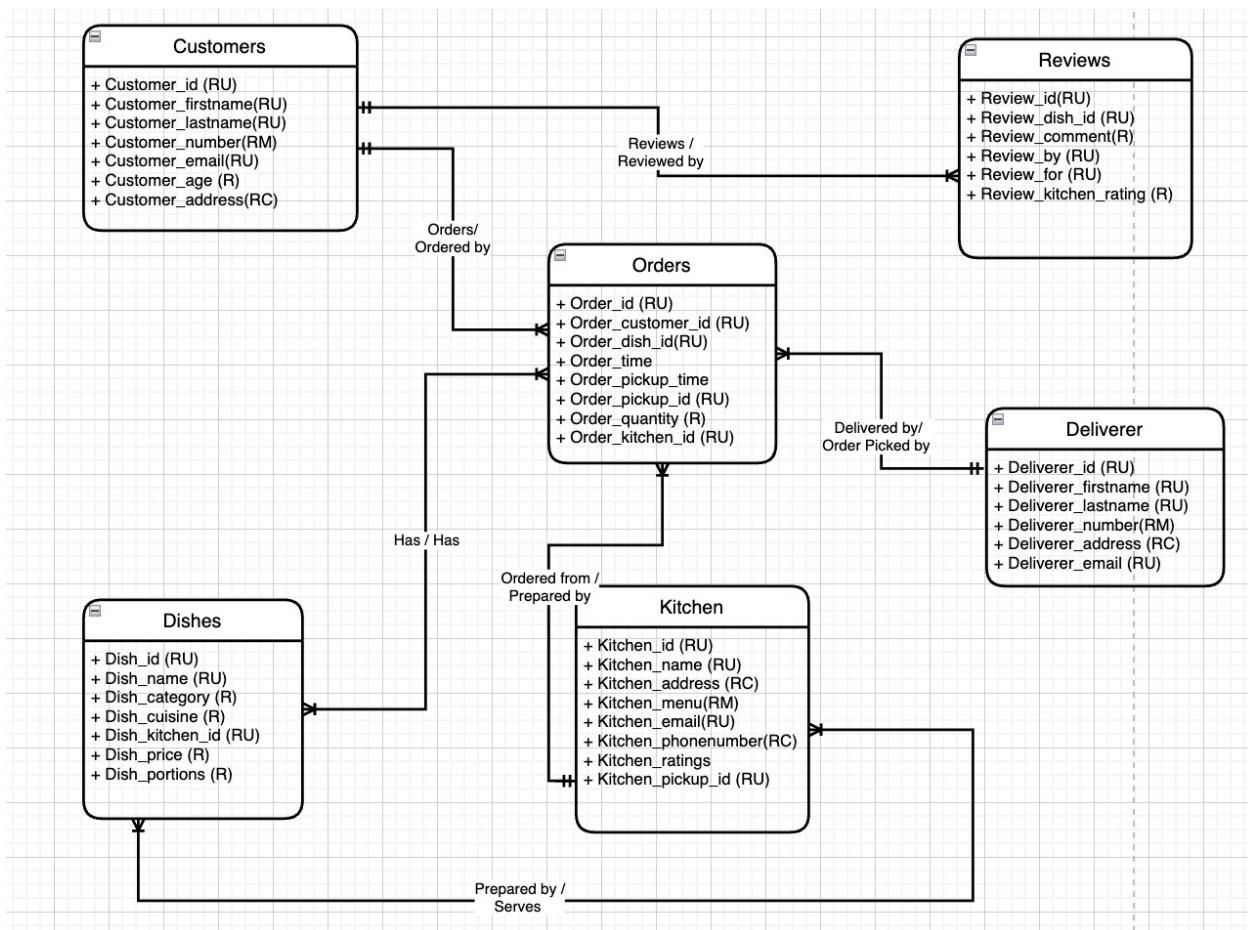
## **BUSINESS QUESTIONS:**

- How can we design a robust and scalable database architecture that accommodates the diverse needs of an online food delivery service, including menu management, customer profiles, order history, and delivery tracking?
- What technologies and frameworks should be employed to create a user-friendly and intuitive platform for customers to browse menus, place orders, and track their deliveries in real time?

## **SOLUTION:**

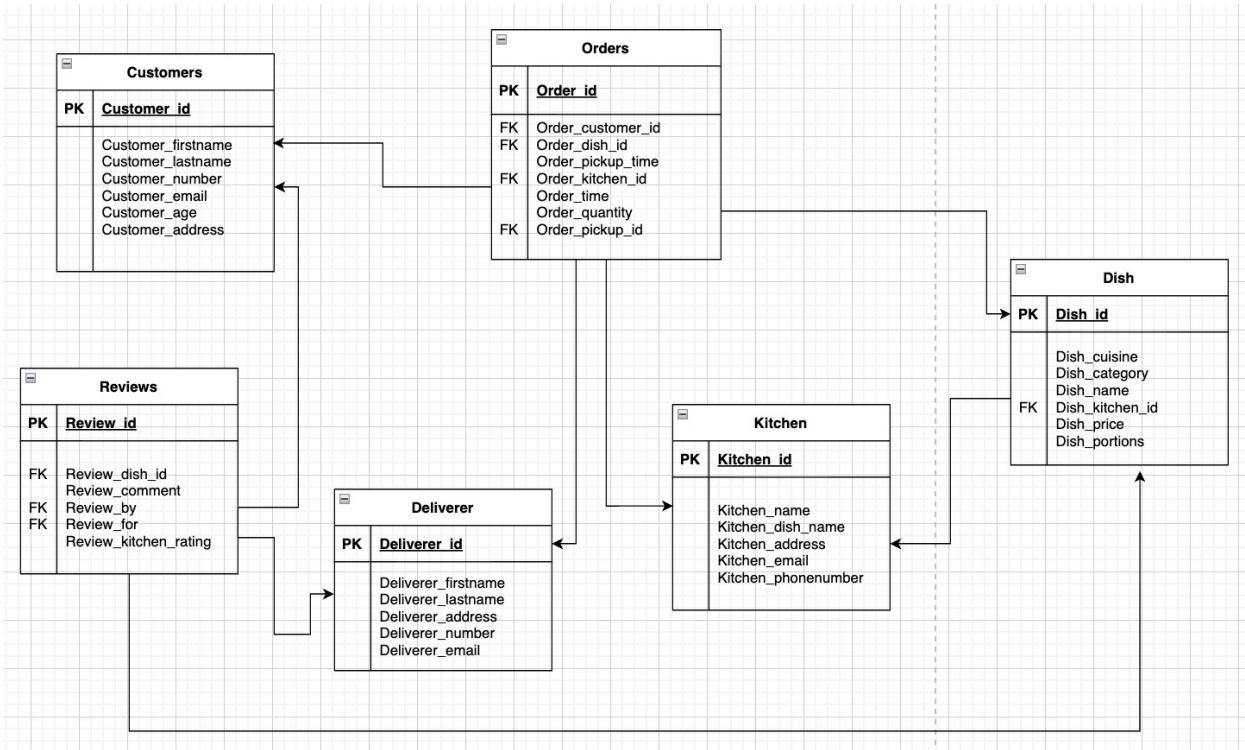
- The project involved a multi-faceted approach, beginning with the design of both conceptual and logical models. These models served as the blueprint for structuring the data ecosystem, ensuring a coherent and organized foundation. Further enhancing the project's depth, I meticulously crafted informative SQL queries tailored for data extraction. This encompassed the creation of parameterized stored procedures, thoughtfully designed indexes, and triggers. These elements collectively streamlined data retrieval processes and fostered efficient data management.
- A pivotal aspect of the project was conducting comprehensive data analysis and profiling using intricate SQL queries across diverse source systems, primarily employing SQL Server. This exploratory analysis sought to unveil hidden insights within the data and extract meaningful patterns. This was achieved through the application of complex SQL queries that navigated the intricacies of the source systems, revealing valuable information for informed decision-making and strategic planning.
- In essence, the project demonstrated a robust mastery of SQL-based techniques, from crafting efficient queries for data extraction to in-depth data analysis. The culmination of these efforts resulted in a well-structured data landscape that enabled data-driven insights, amplifying the project's impact on effective decision-making processes.

## CONCEPTUAL DATA MODEL:



Demonstrated the relationship between different tables in the conceptual model, as well as how each attribute in the table is defined.

## LOGICAL DATA MODEL:



We specified which entity is associated to which other entity in the logical model, as well as all of the constraints of each entity.

## IDENTIFICATION OF DATA LOGIC:

We have implemented data logic on the following as:

- **Customer**: The customer table identifies every customer by a customer ID and lists basic customer information. It also finds the customer information and also update related information.
- **Dishes**: The dish table identifies the dish details from kind of dishes to price/portion of the dish. This table enables the admin to view the information of the dishes a customer orders.
- **Order**: The order table identifies order information a customer makes – duration, quantity.
- **Kitchen**: The kitchen table gives the details of dish maker's information. This is made accessible to customers as well.
- **Deliverer**: To lookup employees job details and also update them who are working at Orange Air
- **Reviews**: The review table provides the rating, commentary information of the dish and kitchen by the customer.

## SQL UP/DOWN SCRIPTS:

**DOWN**

```
1  -- Create the new database if it does not exist already
2  drop DATABASE GhostKitchen
3  IF NOT EXISTS (
4      SELECT [name]
5          FROM sys.databases
6          WHERE [name] = N'GhostKitchen'
7  )
8  CREATE DATABASE GhostKitchen
9  GO
10
11
12 use GhostKitchen
```

```
13 --DOWN
14
15 if exists(select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
16   WHERE CONSTRAINT_NAME='fk_dish_kitchen_name')
17   ALTER TABLE dish DROP CONSTRAINT fk_dish_kitchen_name
18 DROP TABLE if EXISTS dish
19
20 if exists(select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
21   WHERE CONSTRAINT_NAME='fk_orders_dish_dish_id')
22   ALTER TABLE orders DROP CONSTRAINT fk_order_dish_dish_id
23 if exists(select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
24   WHERE CONSTRAINT_NAME='fk_order_order_kitchen_id')
25   ALTER TABLE orders DROP CONSTRAINT fk_order_order_kitchen_id
26 if exists(select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
27   WHERE CONSTRAINT_NAME='fk_order_customer_customer_id')
28   ALTER TABLE orders DROP CONSTRAINT fk_order_customer_customer_id
29 if exists(select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
30   WHERE CONSTRAINT_NAME='fk_orders_deliverer_pickup_id')
31   ALTER TABLE orders DROP CONSTRAINT fk_orders_deliverer_pickup_id
32 DROP TABLE if EXISTS orders
33
34 if exists(select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
35   WHERE CONSTRAINT_NAME='fk_review_review_by')
36   ALTER TABLE reviews DROP CONSTRAINT fk_review_review_by
37 if exists(select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
38   WHERE CONSTRAINT_NAME='fk_review_review_for')
39   ALTER TABLE reviews DROP CONSTRAINT fk_review_review_for
40 if exists(select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
41   WHERE CONSTRAINT_NAME='fk_review_review_deliverer_rating')
42   ALTER TABLE reviews DROP CONSTRAINT fk_review_review_deliverer_rating
43
44 DROP TABLE if EXISTS reviews
45 DROP TABLE if EXISTS deliverer
46 DROP TABLE if EXISTS customers
47 DROP TABLE if EXISTS kitchen
48
```

## UP Metadata

---

```
49  -- UP Metadata
50
51 CREATE TABLE customers (
52     customer_id INT UNIQUE IDENTITY not null,
53     customer_first_name varchar(30) not null,
54     customer_last_name  varchar(30) not null,
55     customer_address   varchar(50) not null,
56     customer_phone_number int not null,
57     customer_email    varchar(20) UNIQUE not null,
58     customer_age      INT not null
59     Constraint pk_customers_customer_id PRIMARY KEY(customer_id)
60 )
61 ALTER table customers alter COLUMN customer_phone_number bigint not null
62
63 CREATE TABLE Kitchen (
64     kitchen_id INT UNIQUE IDENTITY NOT NULL,
65     kitchen_name varchar(100) UNIQUE NOT NULL,
66     kitchen_dish_name varchar(100) not null,
67     kitchen_address  varchar(50) NOT NULL,
68     kitchen_email   varchar(30) NOT NULL,
69     kitchen_phonenumber varchar(10) NOT NULL,
70     Constraint pk_kitchen_kitchen_id PRIMARY KEY(kitchen_id)
71 )
72
73
74 /*CREATE TABLE location (
75     location_id int not null,
76     location_address varchar(50) not null,
77     location_city varchar(25) not null,
78     location_state VARCHAR(25) not null,
79     location_country VARCHAR(10) not null,
80     constraint pk_location_location_id PRIMARY KEY(location_id)
81 ); */
82
83 CREATE TABLE Deliverer(
84     deliverer_id INT UNIQUE IDENTITY NOT NULL,
85     deliverer_first_name varchar(30) NOT NULL,
86     deliverer_last_name varchar(30) NOT NULL,
87     deliverer_number varchar(10) NOT NULL,
88     deliverer_address varchar(50) NOT NULL,
89     deliverer_email  varchar(30) NOT NULL,
90     Constraint pk_deliverer_deliverer_id primary key(deliverer_id),
91 )
```

```

92
93 CREATE TABLE Dish (
94     dish_id INT unique IDENTITY NOT NULL,
95     dish_name varchar(50) NOT NULL,
96     dish_cuisine varchar(50) NOT NULL,
97     dish_category varchar(50) NOT NULL,
98     dish_kitchen_id INT not null,
99     dish_price SMALLINT NOT NULL,
100    dish_portions INT NOT NULL,
101    Constraint pk_dish_dish_id PRIMARY KEY (dish_id),
102    Constraint fk_dish_kitchen_id FOREIGN KEY(dish_kitchen_id) REFERENCES kitchen(kitchen_id)
103 );
104
105
106 CREATE TABLE Reviews (
107     review_id int unique identity not null,
108     review_dish_id INT,
109     review_comment varchar(50) NOT NULL,
110     review_by INT NULL,
111     review_for INT NULL,
112     review_deliverer_rating INT NOT NULL,
113     Constraint pk_reviews_review_id primary key(review_id),
114     Constraint fk_review_review_by FOREIGN KEY(review_by) REFERENCES customers(customer_id),
115     Constraint fk_review_review_for FOREIGN KEY(review_for) REFERENCES kitchen(kitchen_id),
116     Constraint fk_review_review_deliverer_rating FOREIGN KEY(review_deliverer_rating) REFERENCES deliverer(deliverer_id)
117 )
118
119 CREATE TABLE Orders (
120     order_id int IDENTITY not null,
121     order_dish_id INT NOT NULL,
122     order_customer_id INT NOT NULL,
123     order_time FLOAT NULL,
124     order_quantity SMALLINT NOT NULL DEFAULT 1,
125     order_kitchen_id int NOT NULL,
126     order_pickup_time FLOAT NOT NULL,
127     order_pickup_id INTEGER NOT NULL,
128     Constraint pk_orders_order_id PRIMARY KEY(order_id),
129     Constraint fk_orders_dish_dish_id foreign key(order_dish_id) REFERENCES dish(dish_id),
130     Constraint fk_order_order_kitchen_id FOREIGN KEY(order_kitchen_id) REFERENCES kitchen(kitchen_id),
131     Constraint fk_order_customer_customer_id FOREIGN KEY(order_customer_id) REFERENCES customers(customer_id),
132     Constraint fk_orders_deliverer_pickup_id foreign key(order_pickup_id) REFERENCES deliverer(deliverer_id)
133 )
134

```

## UP Data

```
135 --UP Data
136 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
137 VALUES ('Hannah','Baker', '2230 NORTHSIDE, APT 11, ALBANY, NY', 8080367290, 'hbaker@gmail.com', 19);
138 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
139 VALUES ('LINDA','GOODMAN', '731 Fondren, Houston, TX', 4356789345, 'lgoodman@ams.com', 20);
140 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
141 VALUES ('JOHNY','PAUL','638 Voss, Houston, TX',9834561995, 'jpaul@gmail.com', 17);
142 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
143 VALUES ('JAMES','BOND','3321 Castle, Spring, TX',9834666995, 'jbond@gmail.com', 43);
144 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
145 VALUES ('SHERLOCK','HOLMES','123 TOP HILL, SAN Francisco,CA',8089654321,'sholmes@gmail.com', 50);
146 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
147 VALUES ('SHELDON','COOPER','345 CHERRY PARK, HESSE,GERMANY',1254678903, 'scooper@gmail.com', 19);
148 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
149 VALUES ('RAJ','SHARMA','345 FLOYDS, MUMBAI,INDIA',4326789031,'rsharma@gmail.com', 18);
150 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
151 VALUES ('RAJ','SHARMA','345 FLOYDS, MUMBAI,INDIA',4326789031,'rssharma@gmail.com', 16);
152 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
153 VALUES ('SHUBHAM','GUPTA','567 CHANDANI CHOWK, DELHI, INDIA',8566778890, 'sgupta@gmail.com', 53);
154 INSERT INTO customers (customer_first_name, customer_last_name, customer_address, customer_phone_number, customer_email, customer_age)
155 VALUES ('PRATIK','GOMES','334 VITRUVIAN PARK, ALBANY, NY',4444678903, 'pgomes@gmail.com', 18);
156
157
158 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
159 VALUES ('KBs Kitchen','Dal Makhni','Westcott','kb@gmail.com','9090909090');
160 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
161 VALUES ('AKs Kitchen','Pav Bhaji','South Beech','ak@gmail.com','9090109090');
162 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
163 VALUES ('Kzs Kitchen','Pasta','Summer','kzs@gmail.com','9033909090');
164 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
165 VALUES ('Hss Kitchen','Noodles','Genesse','hs@gmail.com','9190909090');
166 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
167 VALUES ('SPs Kitchen','Pani Puri','Eastcott','sp@gmail.com','9009090770');
168 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
169 VALUES ('SSs Kitchen','Aloo Paratha','Chennai St','ss@gmail.com','9095909090');
170 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
171 VALUES ('JCs Kitchen','Chole','Cherry St','jc@gmail.com','9090767690');
172 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
173 VALUES ('SLs Kitchen','Hakka Noodles','Sion','SL@gmail.com','9088909090');
174 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
175 VALUES ('90s Kitchen','Wok Chinese','Southcott','wok@gmail.com','9011111090');
176 INSERT INTO kitchen (kitchen_name,kitchen_dish_name,kitchen_address,kitchen_email,kitchen_phonenumber)
177 VALUES ('K9s Kitchen','Egg Curry','Norstcott','K9@gmail.com','9931523090');
```

```

180 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
181 VALUES ('Vishal','Pandey','3030303030','Euclid Ave','hu@yahoo.com') ;
182 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
183 VALUES ('Jackda','niels','3030303080','Kensington','ja@yahoo.com') ;
184 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
185 VALUES ('Hen','Ishe','33230303030','Redwood','hiu@yahoo.com') ;
186 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
187 VALUES ('Buck','Hardy','3078303030','Heaven Island','bh@yahoo.com') ;
188 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
189 VALUES ('pop','tates','3030909030','Queensland Pl','qp@yahoo.com') ;
190 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
191 VALUES ('Car','Owna','3030309030','Avondale','co@ymail.com') ;
192 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
193 VALUES ('Bus','Light','5030303030','optima st','bl@yahoo.com') ;
194 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
195 VALUES ('Lee','Mon','30303899890','Clarendon','LM@yahoo.com') ;
196 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
197 VALUES ('shal','dey','39030303030','Quiet Pl','sd@yahoo.com') ;
198 INSERT INTO deliverer (deliverer_first_name, deliverer_last_name,deliverer_number,deliverer_address, deliverer_email)
199 VALUES ('Hal','mack','303083030','Euclidson Ave','hm@yahoo.com') ;
200
201
202 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
203 VALUES ('Pav Bhaji','Indian','Breakfast',1,20,10) ;
204 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
205 VALUES ('Pasta','Italian','Lunch',2,30,10)
206 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
207 VALUES ('Dal Makhni','Indian','Dinner',3,20,10)
208 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
209 VALUES ('Noodles','Indo-Chinese','Lunch',4,20,15)
210 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
211 VALUES ('Hakka Noodles','Indian','Dinner',5,20,25)
212 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
213 VALUES ('Pani Puri','Indian','Snack',6,20,9)
214 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
215 VALUES ('Chole','Indian','Lunch',7,20,12)
216 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
217 VALUES ('Aloo Paratha','Indian','Breakfast',8,20,7)
218 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
219 VALUES ('Wok Chinese','Chinese','Meal',9,20,19)
220 INSERT INTO dish ( dish_name, dish_cuisine, dish_category,dish_kitchen_id,dish_price, dish_portions)
221 VALUES ('Egg Curry','Indian','Brunch',10,20,13)

```

```

222
223 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
224 VALUES (1,'Okay',2,1,1) ;
225 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
226 VALUES (2,'Good',1,3,10)
227 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
228 VALUES (3,'Bad',3,2,9)
229 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
230 VALUES (4,'Worse',4,1,8)
231 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
232 VALUES (5,'Average',5,2,7)
233 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
234 VALUES (6,'Very Good',6,4,6)
235 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
236 VALUES (7,'Okay',7,5,5)
237 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
238 VALUES (8,'Good',8,8,4)
239 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
240 VALUES (9,'Okayish',9,6,3)
241 INSERT INTO reviews (review_dish_id,review_comment,review_by, review_for, review_deliverer_rating)
242 VALUES (11,'Worse',10,9,2)
243
244 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
245 VALUES (2,1,0.5,3,1,0.3,1);
246 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
247 VALUES (3,2,1.15,2,2,0.4,2);
248 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
249 VALUES (1,3,0.6,4,3,0.4,6);
250 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
251 VALUES (4,4,0.3,5,4,0.3,4);
252 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
253 VALUES (6,5,1.5,8,5,0.5,5);
254 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
255 VALUES (5,6,0.5,3,6,0.3,6);
256 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
257 VALUES (7,7,2.3,10,7,0.5,7);
258 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
259 VALUES (9,8,0.6,5,8,0.5,5);
260 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
261 VALUES (8,6,2,15,9,0.6,10);
262 INSERT INTO Orders (order_dish_id,order_customer_id,order_time,order_quantity,order_kitchen_id,order_pickup_time,order_pickup_id)
263 VALUES (10,9,0.8,11,7,0.7,10);
264

```

## VERIFICATION

```

265      --Verification
266
267
268      select * from customers
269      select * from Kitchen
270      select * from Deliverer
271      select * from dish
272      select * from reviews
273      select * from Orders
274

```

## QUERIES

```
1
2     Select * FROM Dish
3
4     ---1. Type of Cuisine
5     CREATE VIEW cuisine_type AS
6         SELECT dish_name,dish_category,dish_price
7             FROM Dish
8         WHERE dish_cuisine = 'Indian';
9
10    select * FROM cuisine_type
11
12    drop PROCEDURE change_deliverer_number
13_
14    ---2. Updating Deliverer Phone Number
15
16
17    create PROCEDURE change_deliverer_number (
18        @deliv_id int, @deliv_no varchar(10)
19    ) as begin
20        update deliverer
21            set deliverer_number = @deliv_no
22            where deliverer_id = @deliv_id
23    end
24    ---
25    exec change_deliverer_number
26        @deliv_id = 5,
27        @deliv_no = '9794209845'
28    ---
29    select * FROM Deliverer
30
31
32    drop PROCEDURE change_customer_email
33
```

```

34 ---3. Updating Customer Email
35 create PROCEDURE change_customer_email (
36     @cust_id int, @cust_email varchar(20)
37 ) as begin
38     update customers
39         set customer_email = @cust_email
40         where customer_id = @cust_id
41 end
42
43 ---
44 exec change_customer_email
45     @cust_id = 1,
46     @cust_email = 'pgggggggggggomess@gmail.com'
47 GO
48 ---
49 select * from customers
50
-- 
52
53 ---4. Reviews for all dishes from a Kitchen
54 CREATE VIEW kitchen_review AS
55 SELECT d.dish_name,d.dish_cuisine, r.review_comment
56 FROM Reviews r
57 INNER JOIN kitchen k on k.kitchen_id = r.review_for
58 INNER JOIN dish d on dish_kitchen_id = kitchen_id
59 WHERE k.kitchen_name = 'KBs Kitchen' ;
60
61 SELECT * FROM kitchen_review
62
63

```

```

64
65  ---5. Customer details for all orders more than 3
66  CREATE VIEW orders_more_than AS
67  SELECT c.customer_first_name + ' ' + c.customer_last_name AS customer_name,
68  c.customer_email, c.customer_phone_number, k.kitchen_name
69  FROM Orders o
70  JOIN Customers c ON c.customer_id = o.order_customer_id
71  JOIN Kitchen k ON o.order_kitchen_id = k.kitchen_id
72  WHERE o.order_quantity > 3
73
74  SELECT * FROM orders_more_than
75

76
77  ---6. Kitchen known for? (Customer POV)
78  CREATE PROCEDURE kitch_info (
79      @k_id INT
80  ) AS BEGIN
81      DECLARE @kitch_n VARCHAR (25)
82      DECLARE @kitch_dn VARCHAR (25)
83      DECLARE @kitch_ad VARCHAR (50)
84      SELECT
85          @kitch_n = kitchen_name,
86          @kitch_dn = kitchen_dish_name,
87          @kitch_ad = kitchen_address
88      FROM Kitchen
89      WHERE @k_id = kitchen_id
90      PRINT @kitch_n + ' is known for ' + @kitch_dn + ' and is at ' + @kitch_ad
91  END
92
93  EXEC kitch_info
94      @k_id = 3
95  GO

```

```
create PROCEDURE change_customer_address (
    @cust_id int, @cust_add varchar(50)
) as begin
    update customers
        set customer_address = @cust_add
        where customer_id = @cust_id
end

exec change_customer_address
    @cust_id = 6,
    @cust_add = '644 Westcott Street, Syracuse, NY'
GO
```

## REFERENCES:

- Tutorialspoint
- Geeksforgeeks