USE CASE STUDY REPORT

I. Introduction

The Manager of the restaurant named “Verandah” wants to create a relational database system to map the daily functioning of his restaurant as well as perform statistical analysis to calculate the food item sales of the restaurant. The manager wants a thorough analysis of which customers prefer to dine-in or get online order delivery and which food items have high sales during the hours the restaurant is open. The management also wants to continue with this relational database model if the open multiple restaurants in the future.

The manager mentions the functioning of the restaurant as follows:

The Restaurant has a name, address, phone number, email, opening and closing hours. The restaurant provides multiple delicacies as food items for its customers. The food items have a static cost, name and time of preparation before being served. The customers have a name, phone number and an email ID. A customer can be a dine-in customer or an online customer. A customer can be a dine-in as well as online customer at different times and online customers have an address of delivery.

Dine-in customers get assigned to a particular table in the restaurant and each table has a unique table number as well as the seating capacity of that table. Orders are placed by customers and are uniquely identified by Invoice No. as well as hold the date_time, order type, total amount of the order. Orders are generated differently for dine-in customers and online customers, and a dine-in order cannot be an online order and vice- versa. An order must be a dine-in or an online order. A dine-in order is facilitated by a waiter. An online order has an estimated time of delivery attached to it.

A KOT (kitchen order ticket) is generated for every order, an order can have multiple KOTs generated for it. A KOT holds a request for food items and is uniquely identified by a KOT No. A KOT can have multiple food items and a food item can be present in multiple KOTs.

A bill is generated after the order is fulfilled by the restaurant. This bill holds information about the service tax, food tax or CGST, SGST and the total amount along with taxes of the bill. The bill is paid by the customer.

Scope of Analysis: -

1]The project is based on creating a relational database model to track the sale of food items to different customers and find patterns of which food items are ordered at which time of the day.

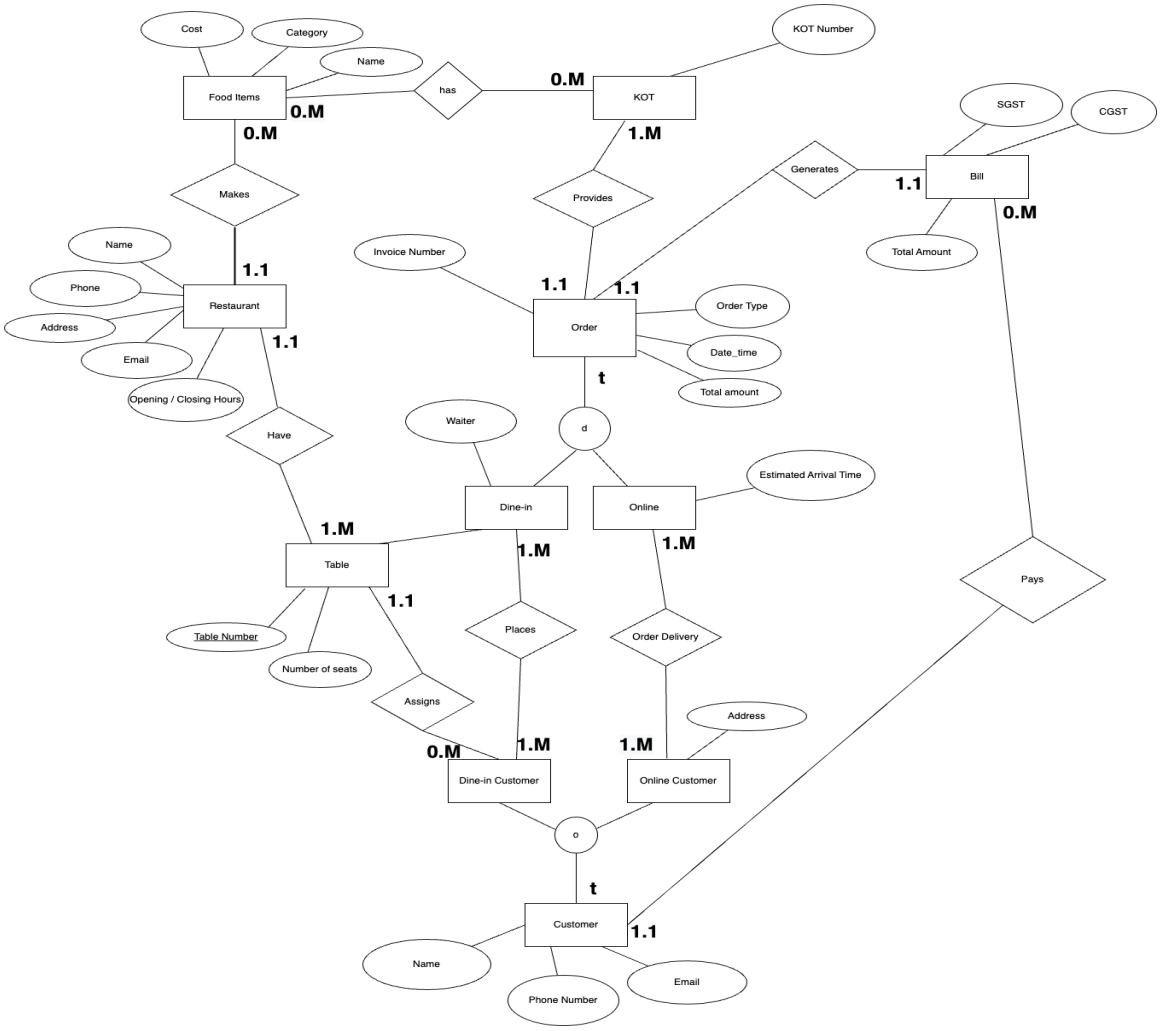
2]The analysis will also be based on which food items are preferred to be eaten as dine-in items and the food items are frequently seen in online and delivery orders. Also, the percentage of customers who prefer to have online orders over dine-in will be part of the analysis.

3]The analysis does not include information about how the employee base of the restaurant is managed by the managers along with their expenses.

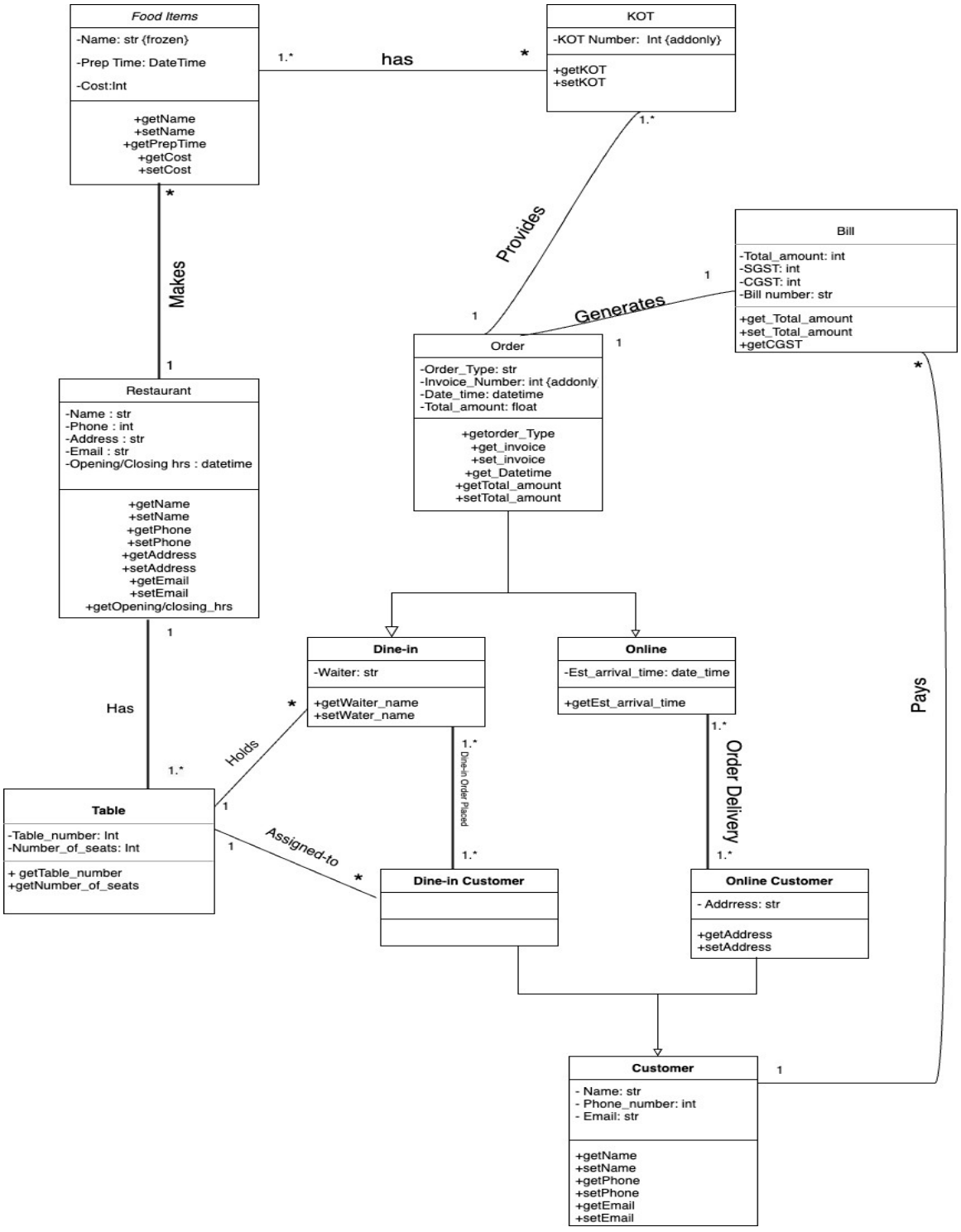
4]The analysis of inventory and supplier details will not be included in the analysis as the manager of the restaurant wants to prioritize food item-based analysis according to his problem statement.

II. Conceptual Data Modeling

EER Diagram:



UML Diagram:-


Food Items
-Name: str {frozen}
-Prep Time: DateTime
-Cost:Int
+getName
+setName
+getPrepTime
+getCost
+setCost
1.*
has
*
KOT
-KOT Number: Int {addonly}
+getKOT
+setKOT
1.*
Provides
1
Makes
*
1
Bill
-Total_amount: int
-SGST: int
-CGST: int
-Bill number: str
+get_Total_amount
+set_Total_amount
+getCGST
1
Generates
1
Order
-Order_Type: str
-Invoice_Number: int {addonly
-Date_time: datetime
-Total_amount: float
+getorder_Type
+get_invoice
+set_invoice
+get_Datetime
+getTotal_amount
+setTotal_amount
*
Pays
Restaurant
-Name : str
-Phone : int
-Address : str
-Email : str
-Opening/Closing hrs : datetime
+getName
+setName
+getPhone
+setPhone
+getAddress
+setAddress
+getEmail
+setEmail
+getOpening/closing_hrs
1
Has
1.*
Dine-in
-Waiter: str
+getWaiter_name
+setWater_name
Online
-Est_arrival_time: date_time
+getEst_arrival_time
*
Holds
1
1.*
Dine-in Order Placed
1.*
1.*
Order Delivery
1.*
Table
-Table_number: Int
-Number_of_seats: Int
+ getTable_number
+getNumber_of_seats
1
Assigned-to
*
Dine-in Customer
Online Customer
- Addrress: str
+getAddress
+setAddress
Customer
- Name: str
- Phone_number: int
- Email: str
+getName
+setName
+getPhone
+setPhone
+getEmail
+setEmail
1

III. Mapping Conceptual Model to Relational Model

Relational Model:-

Primary keys are denoted by Underlined

Foreign keys are denoted by *Italics.*

Restaurant(Name, Address, Phone, Email, Opening/Closing Hours) Customer(Email, Name, phone number)
Order(Invoive Number, Order type, Date_time, Total Amount)

Food Items(Name, Category, cost, *Restaurant Name*) Restaurant Name is a not null foreign key.

KOT(KOT number, *Invoice Number*) Invoice Number is a not null foreign key.

KOT_Holds(*Food Item Name, KOT Number*) Food Item name is a not null foreign key.

Table(Table Number, number of seats, *Restaurant Name*) Restaurant Name is a not null foreign key.

Dine_in_Customer(Email, *Table Number,* Name, Phone Number) Table Number is a not null foreign key.

Dine_in_order(Invoive Number, Order type, Date_time, Total Amount, Waiter, *Table Number*)

Table Number is a not null foreign key.

Dine_in_order_placed(*Invoice Number, Email*) Invoice Number, Email is a not null foreign key. Online Customer(Email, *Table Number,* Name, Phone Number, Address) OnlineOrder(Invoive Number, Order type, Date_time, Total Amount, Estimated Arrival

Time)

Online_Order_Delivery(*Invoice Number, Email*) Invoice Number, Email is a not null foreign key.

Bill(Bill Number, CGST, SGST, Total_Amount, *Invoice Number, Email*) *Invoice Number, Email* are not null foreign keys.

Normalization:-

1NF: The relational model is in 1NF as all values are atomic and single valued so it's in 1NF.

2NF: The relational model does not have any partial dependencies so it's in 2NF.

3NF: The relational model does not have any transitive dependencies so it's in 3NF.

BCNF: As all other attributes are derived from the primary key in the relational model it is in BCNF.

IV. Implementation of Relation Model via MySQL and NoSQL

1) Category-wise topseller : This query will help us in the analysis of which products are the topsellers in each of their categories and why are the ordered with such a high frequency in the restaurant.

```
with cte as(
select i.Category,k.Item_Name, count(k.KOT_ID) as c, dense_rank() over (partition by
i.category order by count(k.KOT_ID) desc) as rank_no from KOT as k inner join Items
as i
on k.Item_Name = i.Item
where i.Category not like '%Extra%'
group by i.Category,k.Item_Name
)
select Category,Item_Name from cte
where rank_no<2
order by c desc;
```

Output:

Category	Itemname
Hot Beverages	The Verandah Filter
Snacks	Sabudana Khichadi
Meals	Chole Bhature
Chat	Spdp
Cold Beverages	Cold Coffee
Salads	Bhajani Thalipeeth
Sandwiches	Peri Peri Fries
Mocktails	Fresh Lime Soda
Deserts	Walnut Halwa
Single Serving Meals	Dal Khichdi

Single Serving Meals	Single Laccha Paratha

2) Online service distribution : This query helps the management analyse how the online orders are delivered. The third-party delivery partner named Swiggy is responsible for majority of the online delivery services. The management can asses what steps have to be taken to make the restaurant delivery system compete on par with the third party delivery system.

```
select name, count(Invoice_no) as number_of_orders
from online_orders
group by name
having  name = 'SWIGGY'
union
(select 'Self_Delivery_System' as name,count(Invoice_no) as number_of_orders
from online_orders
where name != 'SWIGGY');

select min(date),max(date) from online_orders;
```

Output:

name	number_of_orders
Self_Delivery_System	715
SWIGGY	428

3) Days with maximum online orders : This query will help the management analyse which days have the maximum online orders and if those days are days with holidays or festivals.
Also, it can help with the analysis of which days of the week have the maximum online orders.

```
with cte as(
select cast(date as date) as date, count(invoice_No) as total_orders, dense_rank() over
(order by count(invoice_No) desc) as rank_no
from online_orders
group by cast(date as date)
)
select date, total_orders
from cte
where rank_no <6;
```

Output:

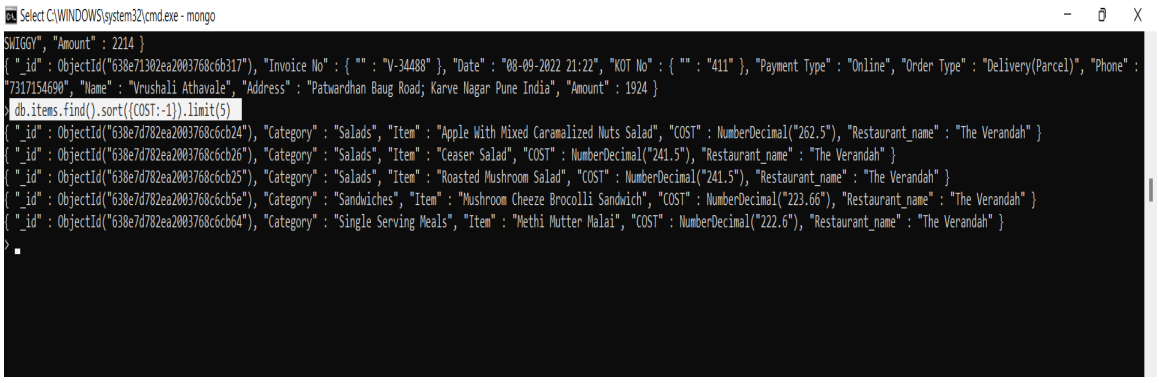
Date	number_of_orders
9/26/22	76
9/4/22	69

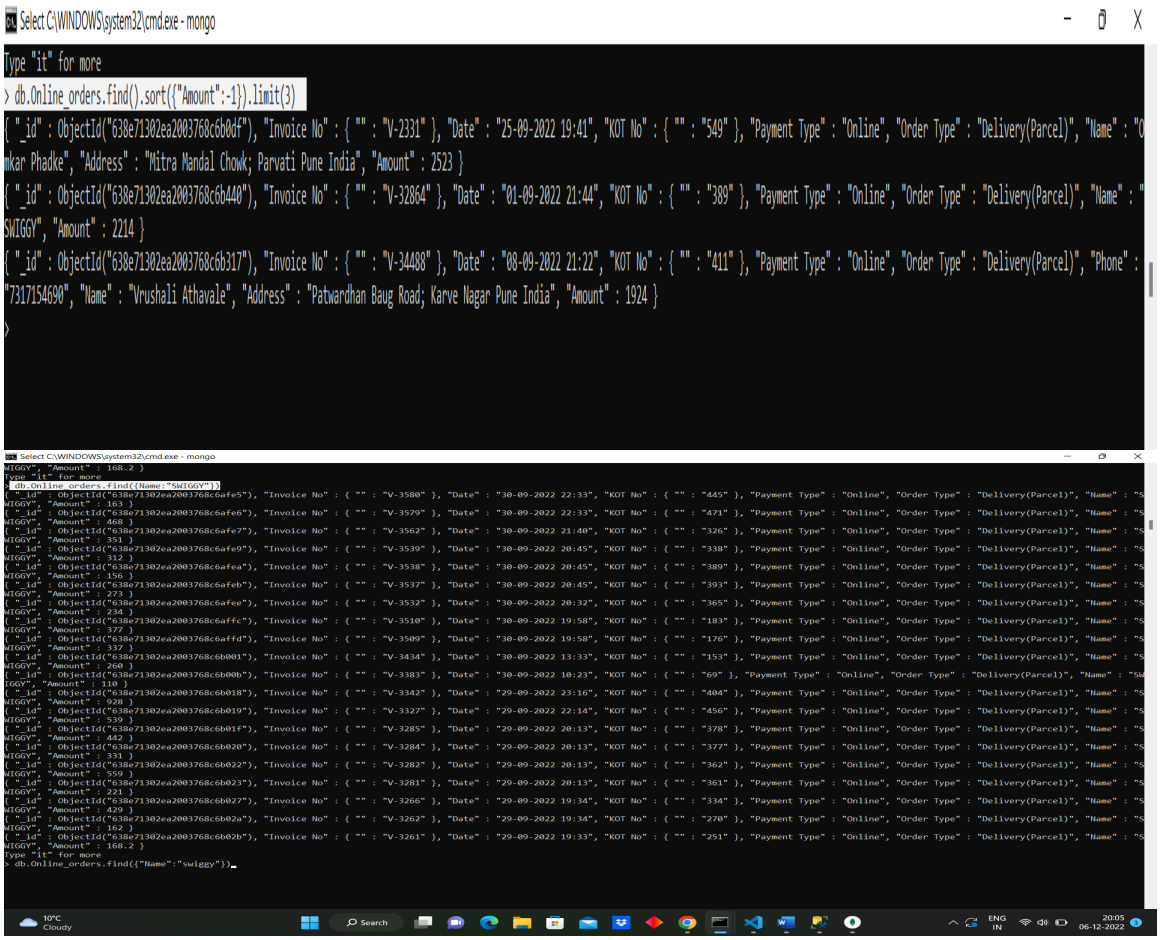
4) Top five selling products : This query helps us understand that which products contribute most to the revenue of the restaurant. Which will help the management further create variations with these products and produce a variety in the menu.

```
with cte as(
select i.item , sum(i.COST)  as revenue,count(k.Qty) as total_quantity, RANK() over
( order by sum(i.COST) desc)as rank_no
from KOT as k inner join items as i
on i.Item = k.Item_Name
group by item
)
select item, revenue, total_quantity from cte
where rank_no < 6;
```

Output:

Item	Revenue	Total_quantity
Chole Bhature	4062.023	21
The Verandah Filter	3465	33
Sabudana Khichadi	3020.089	31
Misal	2640.664	23
Pav Bhaji	2012.856	12




Select C:\WINDOWS\system32\cmd.exe - mongo
Type "it" for more
> db.Online_orders.find().sort({"Amount":-1}).limit(3)
{ "_id" : ObjectId("638e71302ea2003768c6b0df"), "Invoice No" : { "" : "V-2331" }, "Date" : "25-09-2022 19:41", "KOT No" : { "" : "549" }, "Payment Type" : "Online", "Order Type" : "Delivery(Parcel)", "Name" : "O
mkar Phadke", "Address" : "Mitra Mandal Chowk; Parvati Pune India", "Amount" : 2523 }
{ "_id" : ObjectId("638e71302ea2003768c6b440"), "Invoice No" : { "" : "V-32864" }, "Date" : "01-09-2022 21:44", "KOT No" : { "" : "389" }, "Payment Type" : "Online", "Order Type" : "Delivery(Parcel)", "Name" : "
SWIGGY", "Amount" : 2214 }
{ "_id" : ObjectId("638e71302ea2003768c6b317"), "Invoice No" : { "" : "V-34488" }, "Date" : "08-09-2022 21:22", "KOT No" : { "" : "411" }, "Payment Type" : "Online", "Order Type" : "Delivery(Parcel)", "Phone" :
"7317154690", "Name" : "Vrushali Athavale", "Address" : "Patwardhan Baug Road; Karve Nagar Pune India", "Amount" : 1924 }
>