



Restaurant Food Delivery Management System

Presented by Group 3

Name	NUID
Abhinav Manoj Menon	001525815
Fengyi Zhang	001564247
Jayesh Kumar Khattar	001568947
Meghana Sivakumar	001500850
Riddhi Bhatti	001502713

OBJECTIVE AND PROBLEMS ADDRESSED



lack of an organized system to browse the desired cuisines based on ratings and menu and get it delivered home

Welcome to Restaurant Food delivery Management System – a centralized system which helps in ordering food online and helps user in making a food choices based on previous order history

Security and Automation – Credentials of the database users have been encrypted and created automated flows to ease the task of database admin

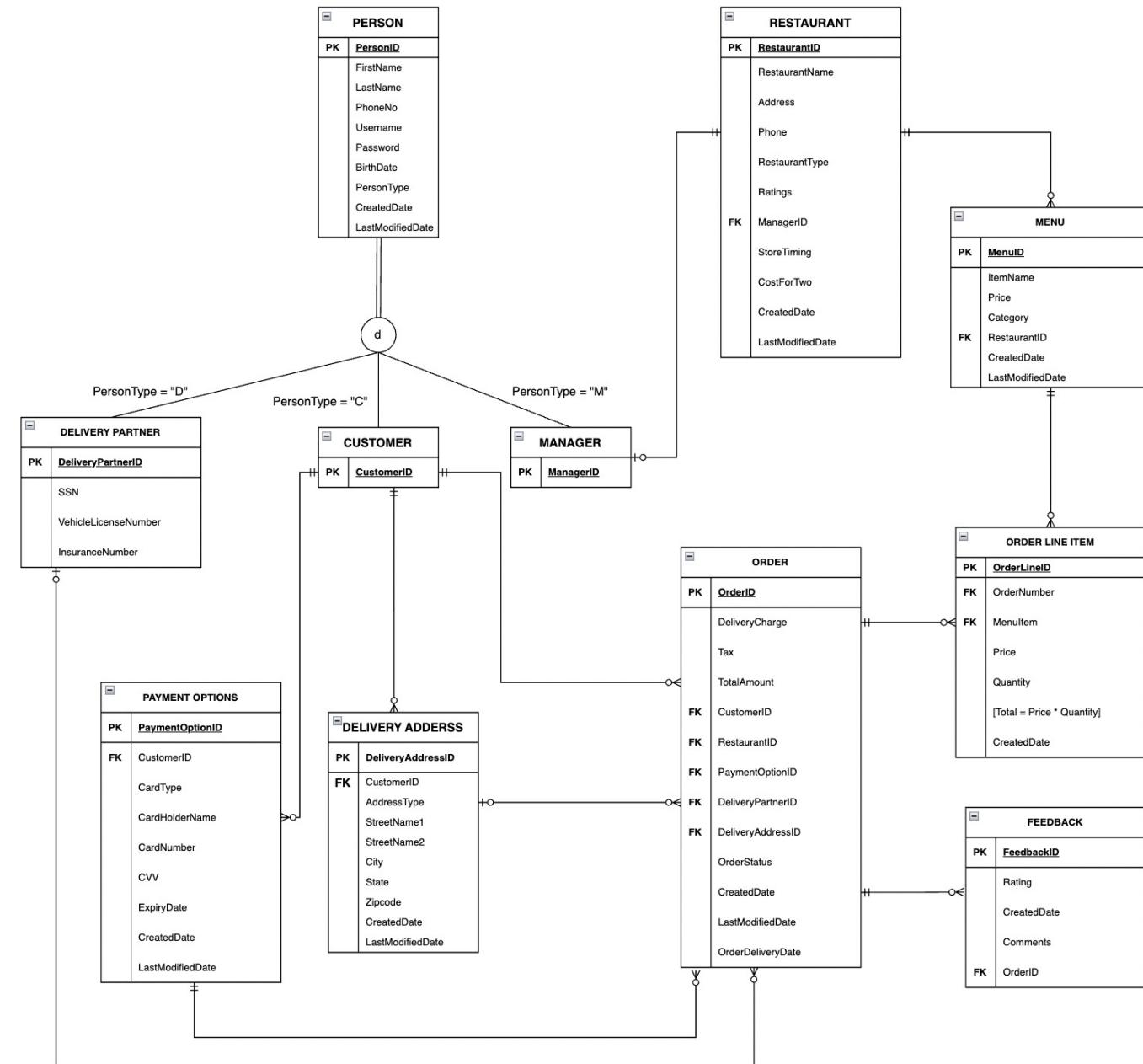
Delivery Tracking – Our system helps in tracking the delivery times of each order and based on this information we can figure out the performance of the delivery partner

Feedback Mechanism – Customer can share the feedback after receiving the order which helps various restaurants to improve their performance and it will also help customers with information of good restaurants.

Data Visualization - Graphs, charts can be used to analyze trends , feedback which helps businesses make critical decision to improve their service.

E-R DIAGRAM

Entity Relationship Diagram
is the absolute key to a
proficient, organized
database design.
It's essential for modeling
the data stored in a
database.



DDL STATEMENTS

Tables

11

Our database system, is made up of the following table and their definitions -

Table Name	PK	FK	Constraints
Person	Yes	No	3
Customer	Yes	Yes	0
[Restaurant Manager]	Yes	Yes	0
[Delivery Partner]	Yes	Yes	4
Restaurant	Yes	Yes	2
[Payment Options]	Yes	Yes	1
[Delivery Address]	Yes	Yes	3
Menu	Yes	Yes	0
[Order]	Yes	Yes(5)	2
[Order Line]	Yes	Yes(2)	0
Feedback	Yes	Yes	0

```

1 IF EXISTS (SELECT name FROM sys.databases WHERE name = N'RestaurantManagement'
2     DROP DATABASE [RestaurantManagement]
3     GO
4
5 CREATE DATABASE [RestaurantManagement];
6 go
7
8 Use RestaurantManagement
9 CREATE MASTER KEY
10 ENCRYPTION BY PASSWORD = 'Damg6210-RestaurantManagement-password';
11 Go
12 CREATE certificate RestaurantManagementPass
13 | WITH SUBJECT = 'RestaurantManagement Password';
14 Go
15
16 CREATE SYMMETRIC KEY PersonPass_SM
17 | WITH ALGORITHM = AES_256
18 | ENCRYPTION BY CERTIFICATE RestaurantManagementPass;
19 Go
20 CREATE SYMMETRIC KEY PaymentPass_SM
21 | WITH ALGORITHM = AES_256
22 | ENCRYPTION BY CERTIFICATE RestaurantManagementPass;
23 Go
24
25 -- Create Person table --
26 USE RestaurantManagement
27 CREATE TABLE Person (
28     PersonID INT IDENTITY(0000,1) PRIMARY KEY,
29     FirstName [VARCHAR](40) NOT NULL,
30     LastName [VARCHAR](40) NOT NULL,
31     Email [VARCHAR](30) NOT NULL,
32     PhoneNo [VARCHAR](10) NOT NULL,
33     Username [VARCHAR](15) NOT NULL,
34     [Password] [varbinary](200) NOT NULL,
35     PersonType [VARCHAR](1) NOT NULL,
36     BirthDate DATE NOT NULL,
37     CreatedDate DATETIME DEFAULT SYSDATETIME(),
38     LastModifiedDate DATETIME,
39     CONSTRAINT chk_PersonType CHECK (PersonType IN ('M', 'C', 'D')),
40     CONSTRAINT chk_Phone CHECK (PhoneNo not like '%[^0-9]%' AND len(PhoneNo)
41     CONSTRAINT chk_Email CHECK (Email like '%[_@%._%]')
42 )
43 go
44 -- Create Customer table --
45 Use RestaurantManagement
46 CREATE TABLE Customer(CustomerID INT NOT NULL,
47 CONSTRAINT [f_CustomerPerson] FOREIGN KEY(CustomerID) REFERENCES Person(PersonID),
48 CONSTRAINT pk_Customer PRIMARY KEY (CustomerID)
49 )
50 Go
51
52 -- Create Delivery Partner table --
53 Use RestaurantManagement
54 CREATE TABLE [Delivery Partner] (
55     DeliveryPartnerID INT PRIMARY KEY,
56     SSN VARCHAR (10) NOT NULL,
57     VehicleLicenseNumber [VARCHAR](19) NOT NULL,
58     InsuranceNo [VARCHAR](10) NOT NULL,
59     CreatedDate DATETIME DEFAULT SYSDATETIME(),
60     LastModifiedDate DATETIME,
61     CONSTRAINT f_DeliveryPerson FOREIGN KEY(DeliveryPartnerID) REFERENCES Person(PersonID),
62     CONSTRAINT chk_Ssn CHECK (SSN not like '%[^0-9]%' AND len(SSN) = 10),
63     CONSTRAINT ck_VehicleLicenseSpecialChar CHECK (VehicleLicenseNumber NOT LIKE '%[^A-Z0-9]%' ),
64     CONSTRAINT ck_InsuranceSpecialChar CHECK (InsuranceNo NOT LIKE '%[^A-Z0-9]%' )
65 )
66 Go
67
68 -- Create Restaurant Manager table --
69 Use RestaurantManagement
70 CREATE TABLE [Restaurant Manager] (
71     ManagerID INT NOT NULL,
72     CONSTRAINT [f_ManagerPerson] FOREIGN KEY(ManagerID) REFERENCES Person(PersonID),
73     CONSTRAINT pk_Manager PRIMARY KEY (ManagerID)
74 )
75 Go
76
77 -- Create Restaurant table --
78 Use RestaurantManagement
79 CREATE TABLE Restaurant(
80     RestaurantID INT IDENTITY(0000,1) PRIMARY KEY,
81     RestaurantName [VARCHAR](40) NOT NULL,
82     RestaurantType [VARCHAR](30) NOT NULL,
83     Rating FLOAT,
84     ManagerID INT NOT NULL,
85     CostForTwo [VARCHAR](2),
86     CreatedDate DATETIME DEFAULT SYSDATETIME(),

```

STORED PROCEDURES

Procedures

6

GetPassword - This SP helps us to retrieve the decrypted password

Use Case – To be used for verifying user while logging them with username and password

Input – password **Eg.** - EXEC **GetPassword** @customerID=2

UpdateOrderStatus - This Stored Procedure helps us to update order status of a particular order

Use Case – To be used for updating order status for an orderID

Input – OrderId, OrderStatus **Eg.** - EXEC UpdateOrderStatus @OrderID=2,@OrderStatus='Delivered';

GetRestaurant - This SP will help us to filter restaurants based on params.

Use Case – to be used to filter restaurants based on filter type and filter value.

Input – FilterType, FilterValue **Eg. 1** - EXEC GetRestaurant @columnname='City',@columnnvalue="Boston",

Eg. 2 EXEC dbo.GetRestaurant @columnname='RestaurantType',@columnnvalue="Indian"

UpdateOrderAmount- This SP helps us to update order amount from all the order line items.

Use Case – Use to calculate total order amount with the help of User Defined Function.

Input – Order Id **Eg.** - EXEC UpdateOrderAmount @OrderID=2, @OrderTotal = @OrderTotal output;

GetOrderHistory - This SP helps us to retrieve the order history of a particular customer.

Use Case – To be used for fetching order history of a customer.

Input – CustomerId **Eg.** - EXEC GetOrderHistory 0

GetPaymentOptions - This SP helps us to payment options of a customer.

Use Case – to be used when fetching payment options of a customer.

Input – CustomerID **Eg.** - EXEC dbo. GetPaymentOptions @CustomerID=10

USER DEFINED FUNCTION

GetOrderAmount -

- ❖ This UD function is called from a procedure-> UpdateOrderAmount
- ❖ Using this, we calculate the **total order amount** on Order using summation of price and quantity columns from Order Line and the amount based on OrderID

Computed Column on UDF

1

User Defined Function

1

VIEWS

Views

3

getRestaurantProfit - This view helps us to identify the profits earned by each restaurant

OrderFeedback - This view helps us to view feedbacks for orders by all customers

Order_placed - This view helps us to identify the no. of orders placed with each restaurant

- **After Insert on Person** [Create Person according to Profile selected]

Automatically creates Customer And Manager based on person Type selected.

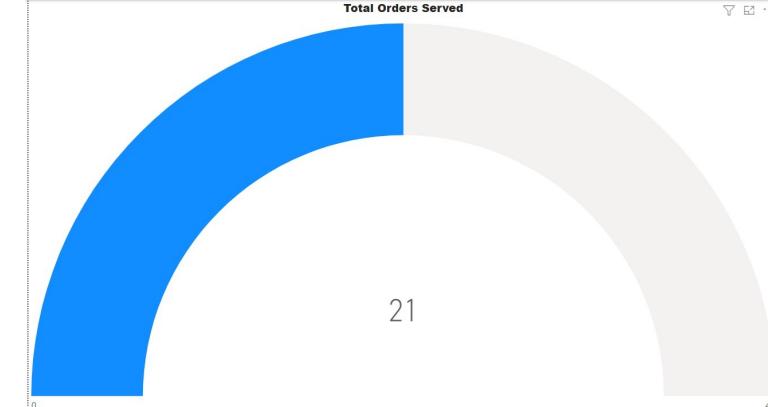
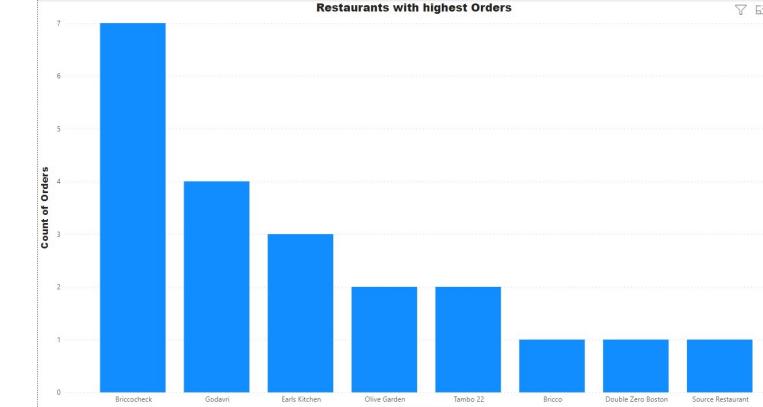
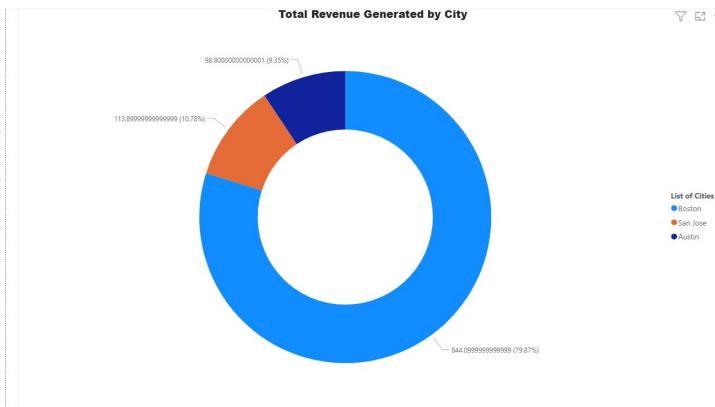
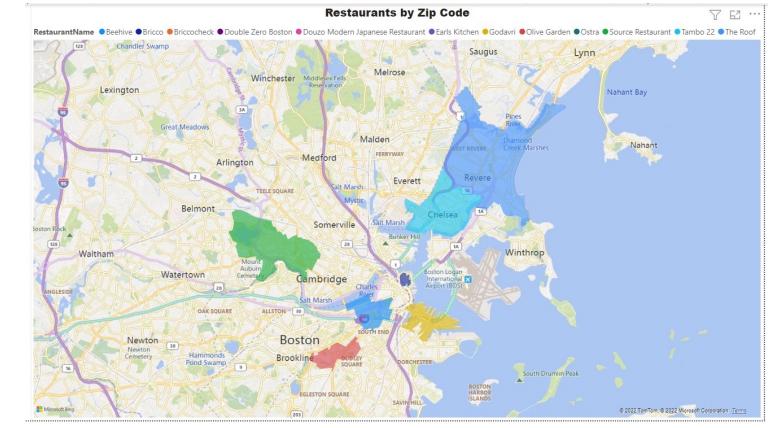
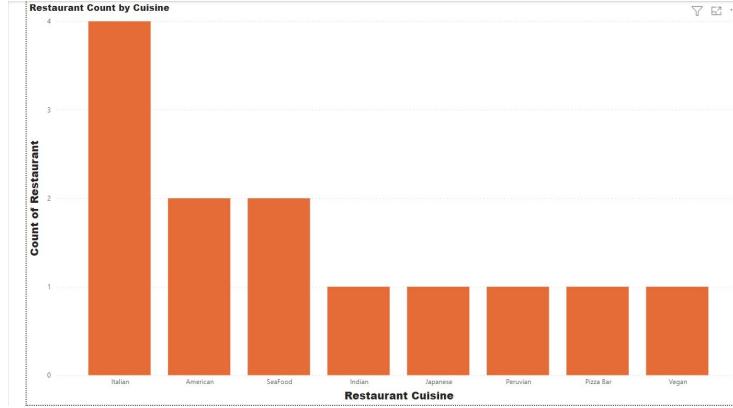
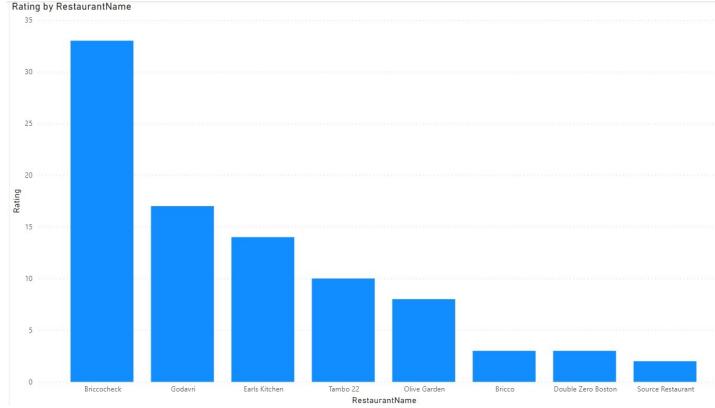
- **After Update Trigger on Order** [Update order Delivery date on Order]

Automatically updates today's order Delivery date on Order after order is delivered

- **After Update Trigger on Person, Restaurant, Order** [Update Last modified date]

Automatically updates last modified date on the records in order to use it while auditing for any record in any of the above tables.

POWER BI VISUALIZATION



GUI

Customer Login Screen -



Delivery Partner Screen -

The Delivery Partner Screen has a blue header with the text "Welcome Delivery Partner". It includes a dropdown menu for "Out of Delivery" set to "Out of Delivery" and a "Change Status" button. Below is a table with columns: OrderID, Name, Order Status, StreetName, State, and ZipCode. Two rows are shown: OrderID 5 for FengyiZhang (In Progress, 150 River Ed..., Medford, Massachusetts, 2155) and OrderID 9 for XinZhao (In Progress, 360 Huntingto..., Boston, Massachusetts, 2115). A "Delivered" button is highlighted in the StreetName column of the second row.

OrderID	Name	Order Status	StreetName	State	ZipCode
5	FengyiZhang	In Progress	150 River Ed...	Medford	Massachusetts 2155
9	XinZhao	In Progress	360 Huntingto...	Boston	Massachusetts 2115

The Customer Order Screen shows a table of items with a modal dropdown for "Olive Garden". The table has columns: Menu ID, ItemName, Category, Quantity, and Total Price. The "Olive Garden" dropdown lists various restaurant names. At the bottom are buttons for "Total Order Price" and "Place Order".

Menu ID	ItemName	Category	Quantity	Total Price
0	Salad	Starter		
1	Burrito	Meal	6	45
2	Cilantro Lime ...	Meal		
3	Nuggets	Starter		
4	Panner Makhni	Main Cour...		

Total Order Price

Place Order