

# Restaurant Management and Food Delivery Database System

ARCHIT NIGAM (NU ID: 001045922)

LOKESH BALAJI PARAMESWARAN(NU ID: 001562875)

THAROON KUMAR VISWANATHAN(NU ID: 001585510)

GAVIN DSA (NU ID: 002191633)

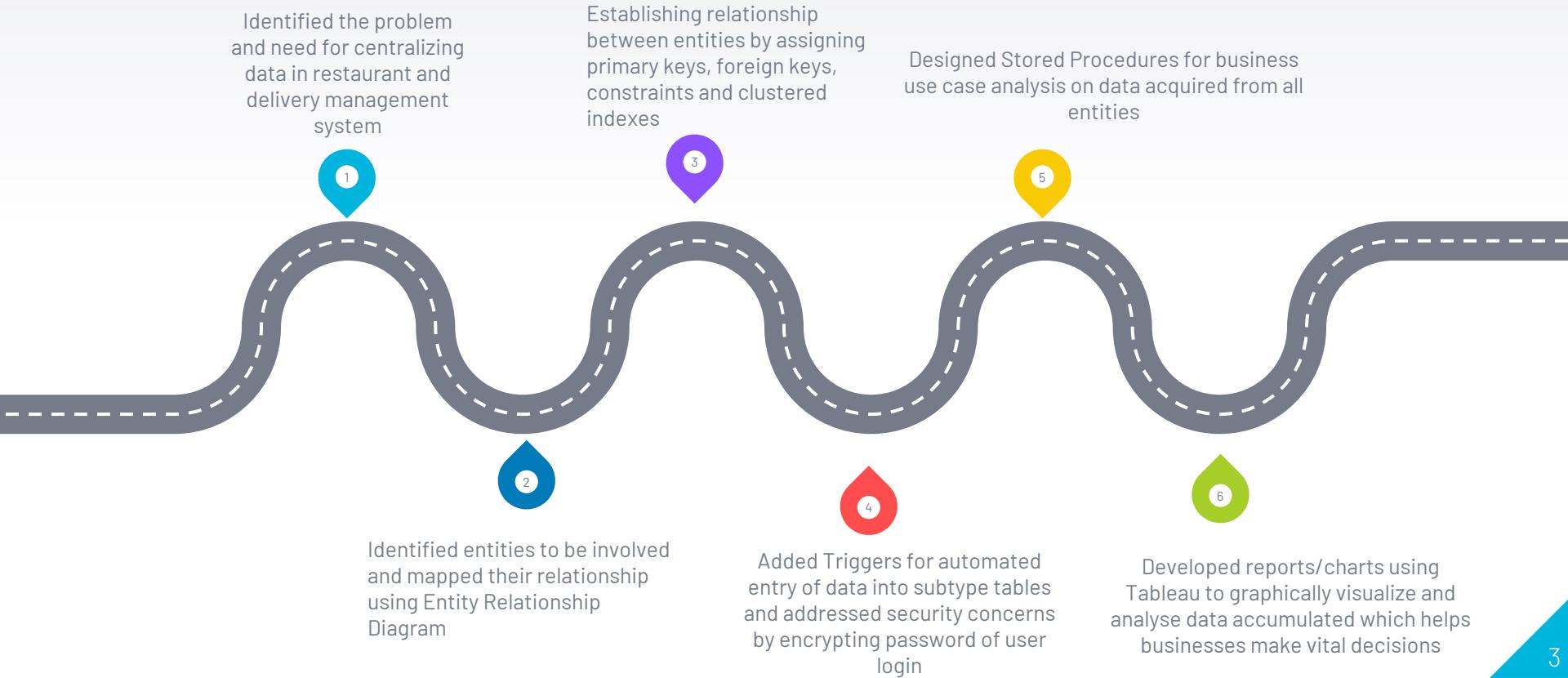


# Problems Identified and Objective

- Lack of centralized data linking the entire restaurant and food delivery entities and workflows together under a common digital ecosystem.
- Ensuring that small and medium level restaurants are granted equal footing and opportunity to compete in culinary and delivery business.
- Auditing and tracking food wastage by restaurants as it is considered an essential tool in the fight against climate change and hunger.



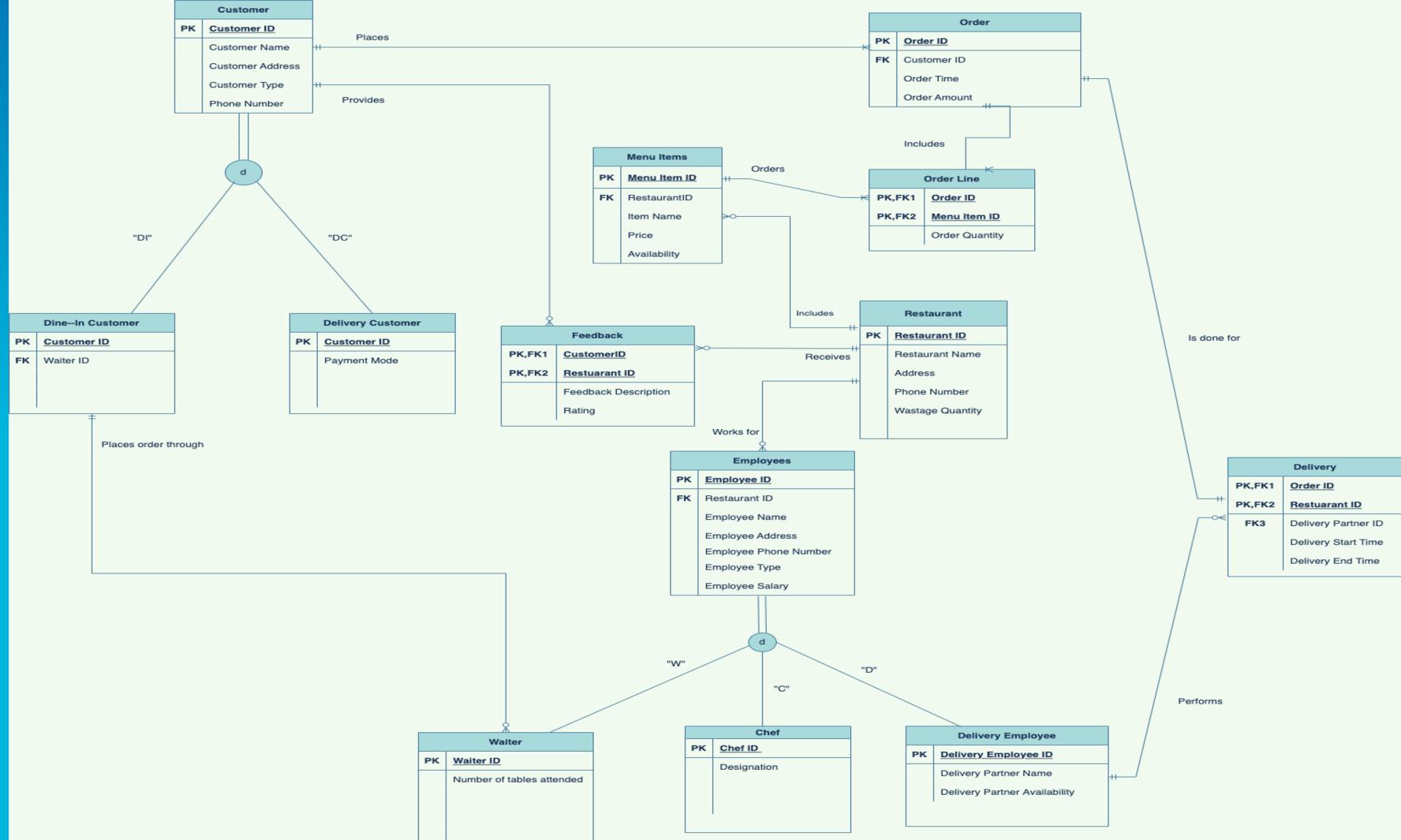
# Roadmap



# E-R Diagram

Entity Relationship Diagram is the absolute key to a proficient, organized database design





# Database Setup

## 1. Run DDL Script

Creates tables, their respective attributes, constraints, clustered and non clustered indexes



## 2. Run Triggers

Helps categorize data automatically into subtype tables with no manual intervention



## 3. Run DML Script

Populates data in all respective tables that can be further used by or for by Stored Procedures, Functions, Views and Data Visualization tools

# Business Use Case #1: Analyze Food Wastage by Restaurant

- Restaurants are required to input their daily food wastage for environmental auditing
- Using views, auditors can be shown the daily wastage and the corresponding restaurant.
- Using Stored Procedures, we can calculate the daily, monthly, half yearly or annual food wastage for the particular restaurant.

The screenshot shows a SQL query window titled 'SQLQuery1.sql - D...JGU8EUN(GAVIN (51))'. It contains T-SQL code to create a stored procedure named 'sp\_CalculateAndDisplayFoodWastageByRestaurants'. The procedure takes a parameter '@rest\_ID int' and uses a SELECT statement to find the top 5 restaurants with their average food wastage quantity in pounds. The results are displayed in a table with columns 'RestaurantName' and 'Average Quantity Wasted in Pounds', showing data for 'The Cheesecake Factory'.

RestaurantName	Average Quantity Wasted in Pounds
The Cheesecake Factory	27.281935

The screenshot shows another SQL query window in SSMS. It contains T-SQL code to create or alter a view named 'Restaurant\_Wastage'. The view selects the total weight of food wasted for each restaurant by summing the waste quantity in pounds from the 'RESTAURANTSWASTAGE' table, grouped by restaurant name. Below the view definition, there is a script for the 'SelectTopNRows' command, which selects the top 1000 rows from the 'Restaurant\_Wastage' view. The results are displayed in a table with columns 'RestaurantName' and 'Total\_Weight', showing data for five restaurants: Neptune Oyster, Playa Bowls, Punjab Place, The Cheesecake Factory, and Toro Boston.

RestaurantName	Total_Weight
Neptune Oyster	97.34
Playa Bowls	505.76
Punjab Place	104.42
The Cheesecake Factory	845.74
Toro Boston	223.61

# Business Use Case #2: Calculate Order Amount after tips

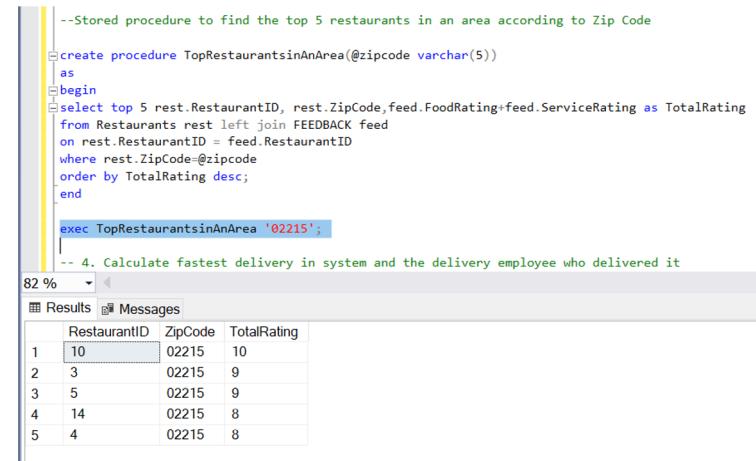
- Stored Procedure provides the provision of manual user input for the tips and adds that amount to the existing sum of individual dishes ordered along with taxes.

The screenshot shows a SQL Server Management Studio window with the following details:

- Query Editor:** The code listed is a stored procedure named `sp_CalculateTotalAmount`. It takes three parameters: `@OrderID int`, `@tips float`, and `@TotalAmount float output`. The procedure first calculates the total amount of items in the `orderline` table by joining it with the `MENUITEMS` table. It then updates the `ORDERS` table to set the `OrderAmount` to the calculated `@TotalAmount` for the specified `OrderID`. Finally, it selects all columns from the `orders` table where `o.OrderID = @OrderID`. A call to the stored procedure is shown at the bottom: `exec sp_CalculateTotalAmount 2, 3.89, @TotalAmount = @TotalAmount;`.
- Results Tab:** The results show a single row of data from the `orders` table. The columns are: OrderID, CustomerID, OrderStatusID, OrderTime, OrderAmount, and RestaurantID. The values are: 1, 2, 4, 2021-11-27 04:22:05.967, 63.89, and 2 respectively.

# Business Use Case #3: Find Top 5 highly rated restaurants in a zip code

- This stored procedure calculates the average rating of a restaurant depending on the food and service rating provided by each customer who has visited the restaurant. Post calculation, it displays the top 5 restaurants in the input zip code.



The screenshot shows a SQL Server Management Studio window. The top pane contains a script for a stored procedure named 'TopRestaurantsinAnArea'. The procedure takes a zip code as input and selects the top 5 restaurants based on their average rating. The bottom pane shows the results of executing the procedure with the input '02215', displaying a table with columns 'RestaurantID', 'ZipCode', and 'TotalRating'.

```
--Stored procedure to find the top 5 restaurants in an area according to Zip Code
create procedure TopRestaurantsinAnArea(@zipcode varchar(5))
as
begin
select top 5 rest.RestaurantID, rest.ZipCode,feed.FoodRating+feed.ServiceRating as TotalRating
from Restaurants rest left join FEEDBACK feed
on rest.RestaurantID = feed.RestaurantID
where rest.ZipCode=@zipcode
order by TotalRating desc;
end

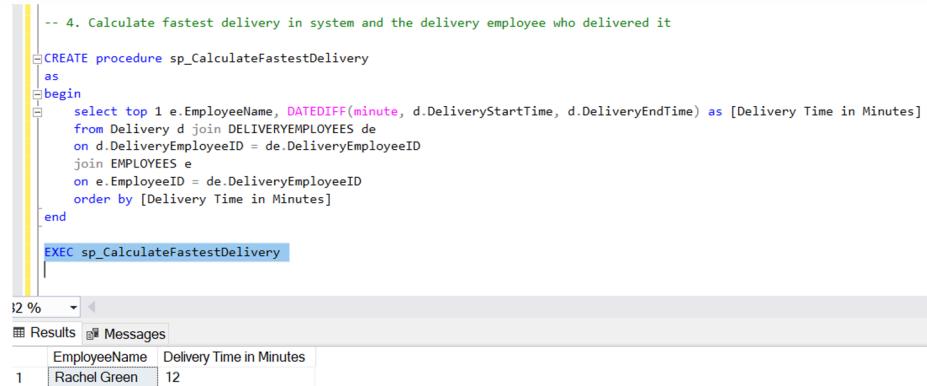
exec TopRestaurantsinAnArea '02215';

-- 4. Calculate fastest delivery in system and the delivery employee who delivered it
```

	RestaurantID	ZipCode	TotalRating
1	10	02215	10
2	3	02215	9
3	5	02215	9
4	14	02215	8
5	4	02215	8

# Business Use Case #4: Fastest Delivery

- This stored procedure calculates the fastest delivery made in the database.



-- 4. Calculate fastest delivery in system and the delivery employee who delivered it

```
CREATE procedure sp_CalculateFastestDelivery
as
begin
    select top 1 e.EmployeeName, DATEDIFF(minute, d.DeliveryStartTime, d.DeliveryEndTime) as [Delivery Time in Minutes]
    from Delivery d join DELIVERYEMPLOYEES de
    on d.DeliveryEmployeeID = de.DeliveryEmployeeID
    join EMPLOYEES e
    on e.EmployeeID = de.DeliveryEmployeeID
    order by [Delivery Time in Minutes]
end

EXEC sp_CalculateFastestDelivery
```

i2 %

Results Messages

	EmployeeName	Delivery Time in Minutes
1	Rachel Green	12

# Business Use Case #5: Delivery Violations

- Late deliveries cause major problems to business and it is essential that they have the option to evaluate and filter any such results.
- The system provides a configurable value (default = 60 minutes) as the threshold for a delivery to be made.

```
74 create procedure sp_ListDeliveryViolations
75 as
76 begin
77
78 select e.EmployeeName, d.DeliveryTimeInMinutes, d
79 from Delivery d join DELIVERYEMPLOYEES de
80 on d.DeliveryEmployeeID = de.DeliveryEmployeeID
81 join EMPLOYEES e
82 on e.EmployeeID = de.DeliveryEmployeeID
83 where d.DeliveryTimeInMinutes > 60
84 end
85
86 EXEC sp_ListDeliveryViolations
```

.21 %

	EmployeeName	DeliveryTimeInMinutes	OrderID
1	Michael Bluth	83	8

# Data Visualization

- Tableau has been used to as the reporting tool for Data Visualization
- Ratings, Food wastage data are displayed on a dashboard for greater visual impact and easier analysis by stakeholders, business managers and analysts.
- The dashboard also displays the list of restaurants that have been involved in these graphs



Customer Count

26

Customer Information

Customer Name	Customer Type
Alexis Rose	DI
Angela Martin	DI
Cameron Tucker	DI
Claire Dunphy	DC
Creed Bratton	DC
David Rose	DC
Dwight Schrute	DI
Erin Hannon	DI
Gloria Delgado	DI
Haley Dunphy	DI
Jan Levinson	DC
Jay Pritchett	DI
Jim Halpert	DI
Joe Pritchett	DI

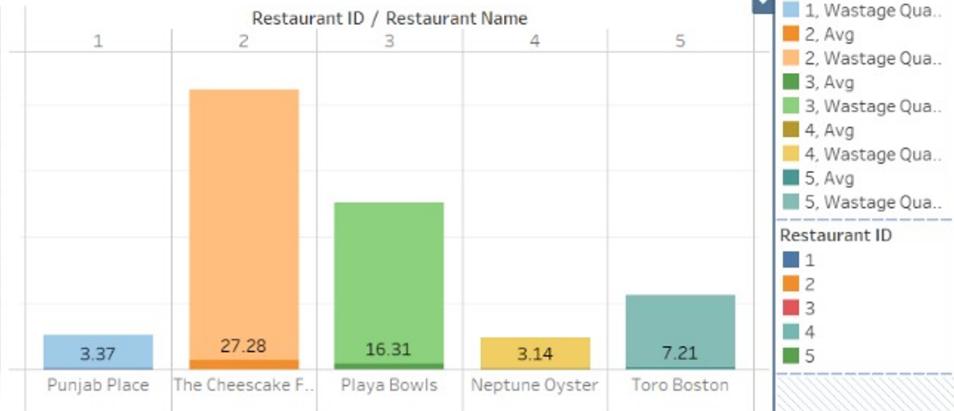
Restaurant Count

15

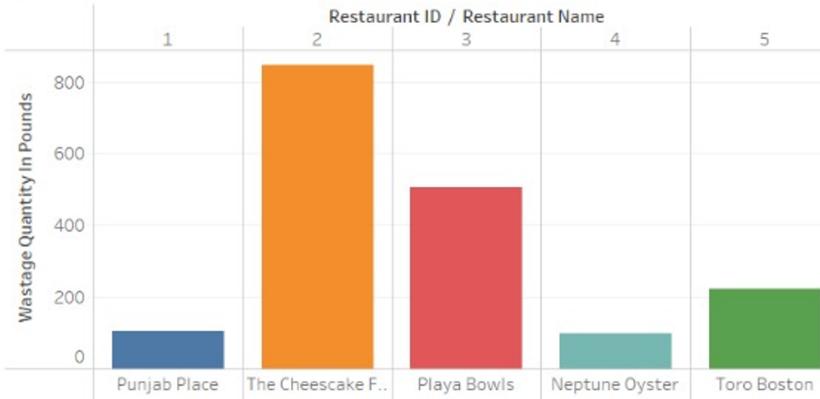
Restaurant Information

Restaurant Name	RestaurantID (..)
B & G Oysters	13
Clio	9
Craigie on Main	12
Menton	14
Mr. Bartleys Gourmet Burgers	11
Neptune Oyster	4
No. 9 Park	7
O Ya	8
Oleana	6
Playa Bowls	3
Punjab Place	1
Sweet Cheeks Q	15
The Cheescake Factory	2
Toro Boston	5
Union Oyster House	10

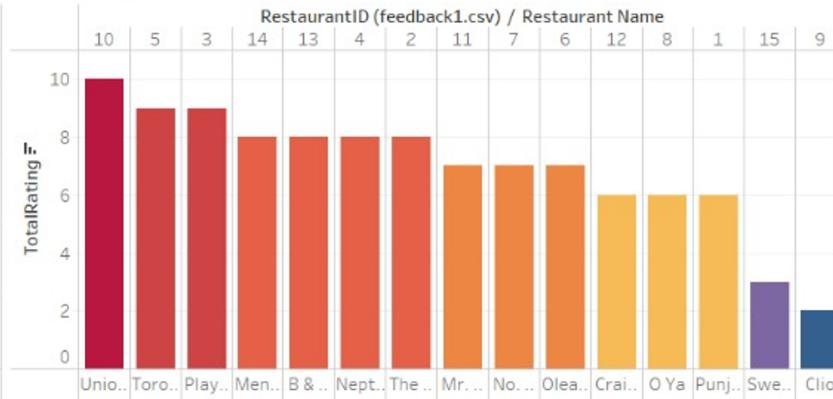
Average to Total Wastage Comparison



Total Restaurant Wastage



Restaurant Ratings



# Other Features

- DDL

```
L USE DMDDFinalProjectDB
E
CREATE TABLE LOCATIONS(
ZipCode varchar(5) not null,
City varchar(30),
CONSTRAINT Locations_PK PRIMARY KEY (ZipCode),
CONSTRAINT LocationsZipCode_CHK CHECK (LEN(ZipCode) = 5)
)

CREATE TABLE RESTAURANTS(
RestaurantID int IDENTITY (1,1) not null,
RestaurantName varchar(30),
StreetAddress varchar(30),
ZipCode varchar(5),
PhoneNumber varchar(10),
Cuisine varchar(20),
Username varchar(20),
[Password] varbinary(400),
CONSTRAINT Restaurants_PK PRIMARY KEY (RestaurantID),
CONSTRAINT Restaurant_FK FOREIGN KEY (ZipCode) REFERENCES LOCATIONS(ZipCode),
CONSTRAINT RESTAURANTSPhoneNumber_CHK CHECK (LEN(PhoneNumber) = 10)
)

CREATE TABLE RESTURANTSWASTAGE(
RestaurantID int not null,
DateOfWastage date DEFAULT getDate() not null,
WastageQuantityInPounds decimal(5,2),
CONSTRAINT RestaurantWastage_PK PRIMARY KEY (RestaurantID, DateOfWastage),
CONSTRAINT RestaurantWastage_FK FOREIGN KEY (RestaurantID) REFERENCES RESTAURANTS(RestaurantID),
INDEX WastageQuantityIndex(WastageQuantityInPounds)
)
```

```
36  CREATE TABLE CUSTOMERS(
37  CustomerID int IDENTITY (1,1) not null,
38  CustomerName varchar(20),
39  StreetAddress varchar(30),
40  Zipcode varchar(5),
41  PhoneNumber varchar(10),
42  CustomerType varchar(2),
43  CONSTRAINT Customers_PK PRIMARY KEY (CustomerID),
44  CONSTRAINT CustomerCustomerType_CHK CHECK(CustomerType IN ('DI', 'DC')),
45  CONSTRAINT CUSTOMERS_FK FOREIGN KEY (Zipcode) REFERENCES LOCATIONS(ZipCode),
46  CONSTRAINT CUSTOMERSPhoneNumber_CHK CHECK (LEN(PhoneNumber) = 10),
47  INDEX CustomerTypeIndex(CustomerType)
48  )

51  )
52
53  CREATE TABLE EMPLOYEES(
54  EmployeeID int not null,
55  RestaurantID int not null,
56  EmployeeName varchar(20),
57  StreetAddress varchar(30),
58  Zipcode varchar(5),
59  PhoneNumber varchar(10),
60  Salary float,
61  EmployeeType varchar(1),
62  CONSTRAINT Employees_PK PRIMARY KEY (EmployeeID),
63  CONSTRAINT Employees_FK1 FOREIGN KEY (RestaurantID) REFERENCES RESTAURANTS(RestaurantID),
64  CONSTRAINT EmployeesEmployeeType_CHK CHECK(EmployeeType IN ('W', 'C', 'D')),
65  CONSTRAINT Employees_FK2 FOREIGN KEY (Zipcode) REFERENCES LOCATIONS(ZipCode),
66  CONSTRAINT EMPLOYEESPhoneNumber_CHK CHECK (LEN(PhoneNumber) = 10),
67  INDEX EmployeeTypeIndex(EmployeeType)
68  )

69  )
70
71  CREATE TABLE WAITERS(
72  WaiterID int not null,
73  Designation varchar(50),
74  CONSTRAINT WAITERS_PK PRIMARY KEY (WaiterID),
75  CONSTRAINT WAITERS_FK FOREIGN KEY (WaiterID) REFERENCES EMPLOYEES(EmployeeID),
76  )
77
78  CREATE TABLE CHEFS(
79  ChefID int not null,
80  Designation varchar(50),
81  CONSTRAINT CHEFS_PK PRIMARY KEY (ChefID),
82  CONSTRAINT CHEFS_FK FOREIGN KEY (ChefID) REFERENCES EMPLOYEES(EmployeeID),
83  )
```

# Other Features

- DDL

```
| CREATE TABLE DELIVERY(
|     OrderID int not null,
|     RestaurantID int not null,
|     DeliveryEmployeeID int,
|     DeliveryStartTime datetime,
|     DeliveryEndTime datetime,
|     DeliveryTimeInMinutes as DATEDIFF(minute, DeliveryStartTime, DeliveryEndTime),
|     CONSTRAINT Delivery_PK PRIMARY KEY (OrderID, RestaurantID),
|     CONSTRAINT Delivery_FK1 FOREIGN KEY (OrderID) REFERENCES ORDERS(OrderID),
|     CONSTRAINT Delivery_FK2 FOREIGN KEY (RestaurantID) REFERENCES RESTAURANTS(RestaurantID),
|     CONSTRAINT Delivery_FK3 FOREIGN KEY (DeliveryEmployeeID) REFERENCES DELIVERYEMPLOYEES(DeliveryEmployeeID)
| )
|
| CREATE TABLE DINEINCUSTOMERS(
|     CustomerID int not null,
|     WaiterID int,
|     CONSTRAINT DINEINCUSTOMERS_PK PRIMARY KEY (CustomerID),
|     CONSTRAINT DINEINCUSTOMERS_FK FOREIGN KEY (WaiterID) REFERENCES WAITERS(WaiterID),
| )
|
| CREATE TABLE DELIVERYCUSTOMERS(
|     CustomerID int not null,
|     PaymentMode varchar(20),
|     CONSTRAINT DELIVERYCUSTOMERS_PK PRIMARY KEY (CustomerID),
| )
|
| CREATE TABLE FEEDBACK(
|     CustomerID int not null,
|     RestaurantID int not null,
|     FeedbackDescription varchar(280),
|     FoodRating int,
|     ServiceRating int,
|     CONSTRAINT Feedback_PK PRIMARY KEY (CustomerID, RestaurantID),
|     CONSTRAINT Feedback_FK FOREIGN KEY (CustomerID) REFERENCES CUSTOMERS(CustomerID),
|     CONSTRAINT FeedbackFoodRating_CHK CHECK(FoodRating IN (1, 2, 3, 4, 5)),
|     CONSTRAINT FeedbackServiceRating_CHK CHECK(ServiceRating IN (1, 2, 3, 4, 5))
| )
|
| CREATE NONCLUSTERED INDEX Feedback_FoodRatingRating_Index ON Feedback(FoodRating ASC);
| CREATE NONCLUSTERED INDEX Feedback_ServiceRating_Index ON Feedback(ServiceRating ASC);
```

# Other Features

- Encryptions

The screenshot shows a database query results window. At the top, there is a code editor with the following SQL query:

```
87  
88 select * from RESTAURANTS  
89  
90
```

Below the code editor is a results grid. The columns are:

RestaurantID	RestaurantName	StreetAddress	ZipCode	PhoneNumber	Cuisine	Username	Password
1	Punjab Place	St Boloph Street	02215	8572618971	Indian	rest1	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
2	The Cheesecake Factory	St Germain Street	02115	8574679894	American	rest2	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
3	Playa Bowls	Queensberry Street	02215	8574109632	Thai	rest3	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
4	Neptune Oyster	Boylston Street	02215	8574196587	Seafood	rest4	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
5	Toro Boston	Newbury Street	02215	8574369679	American	rest5	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
6	Oleana	Irving Street	02145	8572014675	Mediterranean	rest6	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
7	No. 9 Park	60 Forbes Rd	02129	857103213	American	rest7	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
8	O Ya	93 Highland Ave	02147	8579631271	Japanese	rest8	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
9	Clio	199 Ferry St	02148	8574109874	French	rest9	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
10	Union Oyster House	Fenway Park	02225	857106254	Seafood	rest10	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
11	Mr. Barley's Gourmet Burgers	2401 Massachusetts Ave	02115	8579631470	American	rest11	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
12	Craigie on Main	67 Exchange St	02148	8579685555	Italian	rest12	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
13	B & G Oysters	1040 Revere Beach Pkwy	02145	8579782000	Seafood	rest13	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
14	Menton	639 Tremont St	02215	8571003652	Japanese	rest14	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...
15	Sweet Cheeks Q	569 Columbus Ave	02115	8579205478	American	rest15	0x003A80CE26E46C46B9A37A68496B3C770200000F2596C...

# Other Features

- Triggers

```
INSERT INTO CUSTOMERS(CustomerName, StreetAddress, ZipCode, PhoneNumber, CustomerType)
VALUES ('Archit Nigam', '939 Old Bayberry Dr.', '02215', '8578741984', 'DI')

select *from DINEINCUSTOMERS;

CREATE TRIGGER [dbo].[Customer_INSERT]
ON [dbo].[Customers]
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @CustomerType varchar(2)
    DECLARE @CustomerId INT

```

82 %

Results Messages

	CustomerID	WaiterID
1	1	NULL
2	2	NULL
3	4	NULL
4	5	NULL
5	7	NULL
6	8	NULL
7	10	NULL
8	11	NULL
9	13	NULL
10	14	NULL
11	15	NULL

```
SQLQuery3.sql - D...JGU8EU\GAVIN (54) SQLQuery2.sql - D...JGU8EU\GAVIN (52)* X
INSERT INTO EMPLOYEES(EmployeeID, RestaurantID, EmployeeName, StreetAddress, ZipCode,
VALUES(32, 1, 'Gavin Dsa', '52 Primrose Avenue', '02129', '8574985209', 1500, 'W')

select *from EMPLOYEES;
select *from WAITERS;

use DMODFinalProjectDB
CREATE TRIGGER [dbo].[Employees_INSERT]
ON [dbo].[Employees]
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @EmployeeType varchar(2)
    DECLARE @EmployeeId INT

```

82 %

Results Messages

	WaiterID	Designation
1	1	NULL
2	4	NULL
3	7	NULL
4	10	NULL
5	13	NULL
6	16	NULL
7	19	NULL
8	22	NULL
9	25	NULL
10	28	NULL
11	32	NULL

# Achievements

## Common Digital Ecosystem

Data of Restaurants, Customers, Orders, Employees and delivery partners were combined under one umbrella for easier accessibility.

## Data Visualization

Graphs, charts can be used to analyze trends, feedback which helps businesses make critical decisions to improve their service.

## Combating Food Wastage

Climate change can be fought by preventing food wastage, the application provides detailed analysis of wastage by restaurants on daily, monthly and yearly scale

## Delivery Tracking

Food delivery is one of the most competitive markets, the system helps track delivery times for each order and also lists delivery time violations which can help delivery partners improve their performance

## Security and Automation

Credentials of database users have been encrypted and automated workflows have been provided to ease the task of database administrator

## Feedback Mechanism

System has been designed keeping in mind that customers can rate their experience after ordering from a restaurant which helps businesses improve themselves and also can be viewed by prospective visitors

# ► Roles and Responsibilities

## **Archit Nigam**

E-R Diagrams, DDL, DML,  
Encryption

## **Lokesh Balaji**

E-R Diagrams, Tableau Analysis

## **Tharoon Kumar Viswanathan**

E-R Diagrams, Triggers, Views

## **Gavin Dsa**

E-R Diagrams. Stored  
Procedures

# THANKS!

Any questions?

