Tasty Times Food Service

Database Management System

FINAL REPORT

Group 6

Background

Tasty Times Food Service is an On-The Go food truck business that currently sells in three major cities across the United States, namely, Chicago, New York City and Boston. The company has ten trucks in total and is currently owner owned and operated. The owner, Ms. Gartner, has decided that each truck will sell just one item and shall expand to other items in the future. As the business continues to grow, Ms. Gartner is looking to franchise the model and hence, has asked us to create a Database Management System whereby she can seek more clear information about her trucks, employees, products sold and the revenue made.

As the business continues to grow, we have created a SQL program that allows the business to understand the business done on each truck using an identification ID for each truck. This ID helps Tasty Times Food Service' owner to look clearly into what product is being sold best at which location. The business currently sells pizzas, burgers, hotdogs and shakes. These items are fast moving goods, however, if Ms. Gartner plans to expand, she will be looking at items that may take longer to cook since they could lead to a larger profit per item. Moreover, the business will also be considering adding a larger product line up to each truck.

Our role with Tasty Times Food Service is to help her understand using a DBMS that the potential to grow her business is large and that our platform would help her clearly recognize what product shall be expanded to new and other trucks and what product shall be discontinued. Our platform shall also help Ms. Gartner recognize which employee and manager are selling the most and what tips are being given out. This shall help her recognize levels for a bonus and promotion as the business expands. We have also color coded the trucks which shall help the business what product is being sold in which city, and hence, the business for each product can be compared city to city.

These changes shall help Tasty Times Food Service and Ms. Gartner to clearly seek information about her business and make better decisions while considering expansion plans.

Introduction

Mission Statement:

The Database Management System for Tasty Times Food Service aims to bring a clearer look on their business plans and identify key areas of growth and improvement as they seek to move towards a franchise model.

Mission Objectives:

- Design a Database Management System for Tasty Times Food Service that shall help understand the most profitable items
- Understand which city is the most profitable currently for Tasty Times' business
- Seek information on the most bankable employee in each city
- Conduct thorough research on best products as Tasty Times moves to Tier 2 and Tier 3 cities

Data Modeling

Business Scenario:

Tasty Times is a food service company that has a fleet of food trucks located in Indianapolis, IN. Each food has a unique menu offering different flavors of food for every taste bud. The company wants to record information about the fleet of trucks, the employees and the items sold to customers.

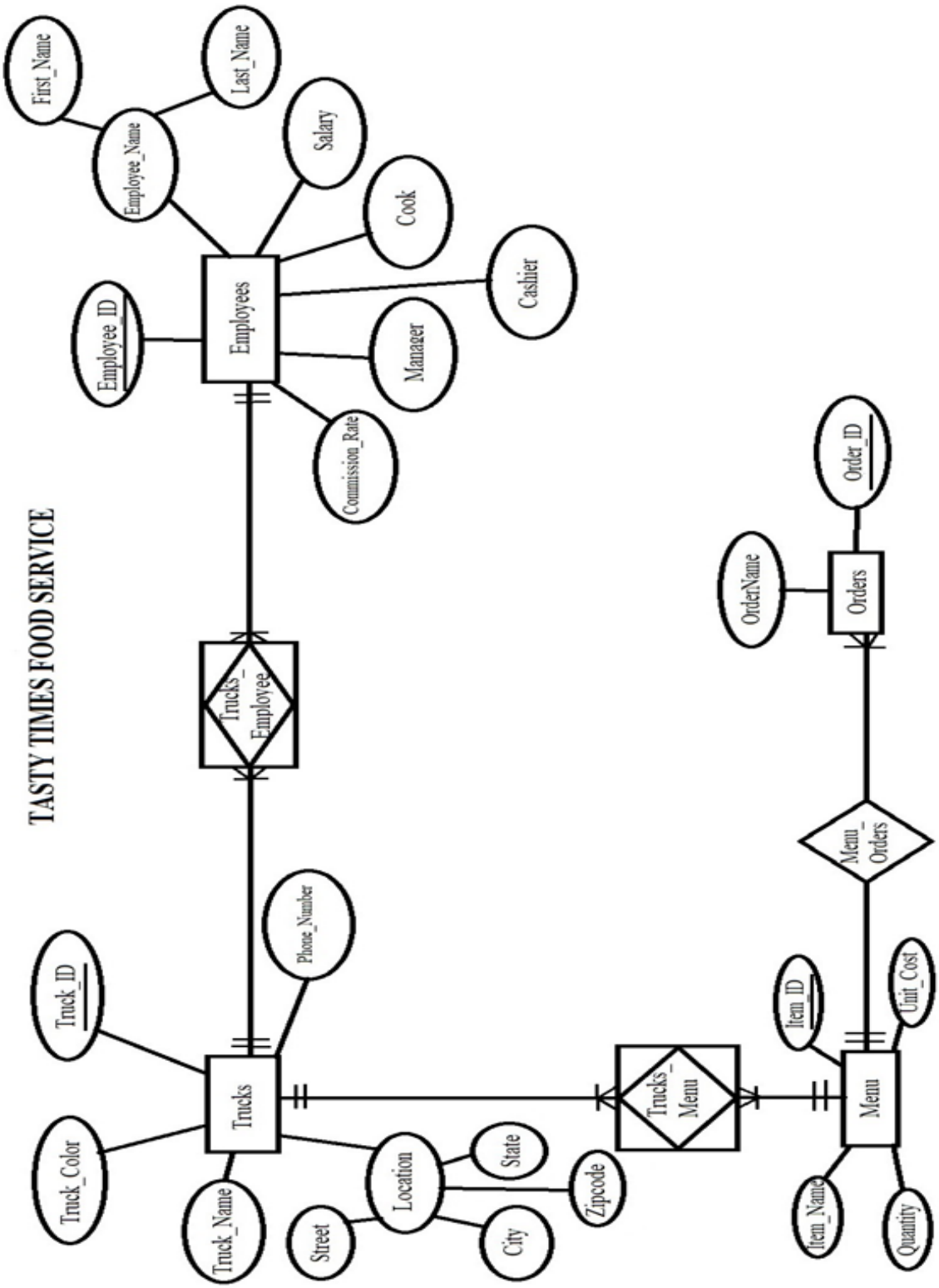
Entity and Attribute:

- For Trucks, the database stores Truck_ID (Primary Key), Truck_Color, Truck_Name, Location composed of Street, City, State and Zipcode, Phone_Number.
- For Employees, the database stores Employee_ID (Primary Key), Employee_Name composed of First_Name and Last_Name, Salary, Commission_Rate, Manager, Cook, Cashier.
- For Menu, the database stores Item_ID (Primary Key), Item_Name, Quantity, Unit_Cost.
- For Orders, the database stores Order_ID (Primary Key), OrderName.

Relationship:

- Tasty Times has a number of trucks. Each truck has one or more menus. Each menu could be served at one or more trucks.
- Each truck is handled by one or more employees and each employee can work in one or more trucks.
- An order is placed for one and only one menu. A menu could have one to many orders.

ER Diagram:


TASTY TIMES FOOD SERVICE
First_Name
Last_Name
Employee_Name
Salary
Cook
Cashier
Employee_ID
Employees
Manager
Commission_Rate
Trucks_
Employee
Truck_ID
Phone_Number
Truck_Color
Trucks
Truck_Name
Location
State
Street
City
Zipcode
Trucks_
Menu
Item_ID
Menu
Unit_Cost
Item_Name
Quantity
Menu_
Orders
OrderName
Orders
Order_ID

Relational Data Model

Relational Schema:

Employees (Employee_ID, First_Name, Last_Name, Salary, Commission_Rate, Manager, Cook, Cashier)

Trucks (Truck_ID, Truck_Color, Truck_Name, Street, City, State, Zipcode, Phone_Number)

Trucks_Employee (Truck_ID, Employee_ID)
FK FK

Orders (OrderID, OrderName, Item_ID)
FK

Menu (Item_ID, Item_Name, Quantity, Unit_Cost)

Truck_Menu (Truck_ID, Item_ID)
FK FK

Normalization:

The Relational Schemas are in 3NF (Full Functional Dependency). The values are atomic, i.e., there are no multi-valued or composite attributes. They do not have any transitive and partial dependencies.

Functional Dependency	Type of Dependency
Employee_ID→First_Name, Last_Name, Salary, Commission_Rate, Manager, Cook, Cashier	Full
Truck_ID→Truck_Color, Truck_Name, Street, City, State, Zipcode, Phone_Number	Full
OrderID→ OrderName, Item_ID	Full
Item_ID→Item_Name, Quantity, Unit_Cost	Full

Data Description:

- **Employees:** The entity has eight attributes with Employee_ID as the primary key: Employee_ID, First_Name, Last_Name, Salary, Commission_Rate, Manager, Cook, Cashier
- **Trucks:** The entity has eight attributes with Truck_ID as the primary key: Truck_ID, Truck_Color, Truck_Name, Street, City, State, Zipcode, Phone_Number
- **Trucks_Employee:** The associative entity has two attributes Truck_ID and Employee_ID. They are both primary and foreign keys
- **Orders:** The entity has three attributes: OrderID, OrderName, Item_ID. OrderID is the primary key and Item_ID is the foreign key.
- **Menu:** The entity has four attributes with Item_ID as the primary key: Item_ID, Item_Name, Quantity, Unit_Cost
- **Truck_Menu:** The associative entity has two attributes Truck_ID and Item_ID. They are both primary and foreign keys

Data Model and Design Choices

As Tasty Times Food Service has grown, so has their need for a database. A database was needed for this client because they have multiple trucks with multiple menus and a growing employee base. In our ERD, we created tables to represent each of the Trucks, Menus, Orders, and Employees. Trucks includes all of the basic information such as Location, Name, Color, Phone number, and the Truck ID. We created an associative entity to connect the Trucks table to Menu. Each truck can have one to many menus and each menu can have one to many trucks. The Menu table holds info like the Item ID, Item Name, Quantity, and Unit Cost. The Menu table has a normal relationship with Orders. A menu can have one to many orders and an order can only be from one and only one menu. This means that each person can only order one item at a time, but each menu item can be ordered by more than one order.

On the top of the ERD, we have created an associate entity relationship between Trucks and Employees. Each truck can have one to many employees and each employee can work at one to many trucks. The reason that we have created associative entities between The Trucks and Employees Tables and the Trucks and Menus Tables is because we want to allow for additional growth in the future. When we use the associative entity between Trucks and Employees, this allows for one to many employees to work at one to many trucks and does not have a restriction as to what truck you need to work for. Additionally, when we use the associative entity between Trucks and Menu, this allows for our menu selection at trucks to grow. Currently, each Truck only sells one item, but the associative entity allows us to easily add additional menu items to additional trucks.

SQL Create Table Script

1. Create Table Script

Employees:

```
Create table Employees (
Employee_ID integer,
First_Name varchar(100),
Last_Name varchar(100),
Salary numeric(7,2),
Commission_Rate numeric(4,2),
Manager varchar(100),
Cook varchar(100),
Cashier varchar(100),
Primary Key (Employee_ID)
);
```

Trucks:

```
Create table Trucks (
Truck_ID integer,
Truck_Color varchar(100),
Truck_Name varchar(100),
Street varchar(100),
City varchar(100),
State varchar(100),
ZipCode integer,
Phone_Number numeric(10),
Primary Key (Truck_ID)
);
```

Trucks_Employee :

```
Create table Trucks_Employee (
Truck_ID integer,
Employee_ID integer,
Primary Key (Truck_ID,Employee_ID)
);
```

Menu:

```
Create table menu (
Item_ID varchar(100),
Item_Name varchar(100),
Quantity int,
Unit_Cost Numeric(3,2),
Primary Key (Item_ID)
);
```

Truck_Menu:

```
Create table Truck_Menu (
Item_ID varchar(100),
Truck_ID varchar(100),
Primary Key (Item_ID, Truck_ID)
);
```

Orders:

```
Create table Orders (
Order_ID varchar(4),
OrderName text,
Item_ID varchar(3),
Primary Key (Order_ID)
);
```

2. Created Tables

Employees:

Employee_ID	First_Name	Last_Name	Salary	Commission_Rate	Manager	Cook	Cashier
11	Jacob	Herro	50000.00	0.25	Yes	No	No
12	James	Tomkins	30000.00	0.15	No	Yes	No
13	Kylie	Kapoor	22000.00	0.05	No	No	Yes
14	Kareena	Jenner	32000.00	0.15	No	Yes	No
15	Rich	Wright	55000.00	0.25	Yes	No	No
16	Mike	Johnson	20000.00	0.05	No	No	Yes
17	Jonathan	Day	38000.00	0.15	No	Yes	No
18	Bob	Odenkirk	24000.00	0.05	No	No	Yes
19	Arturo	Almeida	24000.00	0.05	No	No	Yes
20	Maria	Gomez	21000.00	0.05	No	No	Yes

Trucks:

Truck_ID	Truck_Color	Truck_Name	Street	City	State	Zipcode	Phone_Number
1	Red	Jade	Mass Ave	Chicago	IL	47903	6308807571
2	Blue	Simon	Stoney Creek	Boston	MA	35607	7659908763
3	Black	Rachel	South St	New York	NY	45632	3234568900
4	Red	Karen	Rider Dr	Chicago	IL	90876	6309897238
5	Blue	Carter	Villa Park	Chicago	IL	90210	6306567433
6	Blue	Kit	Old Trafford	New York	NY	85002	3239098790
7	Blue	Corey	Etihad Dr	Boston	MA	54667	7652239084
8	Black	Dez	Gold Park	New York	NY	32334	3236678909
9	Red	Dwayne	Al Riwaz	Chicago	IL	12345	6308874221

Truck_Employee:

Truck_ID	Employee_ID
1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19
10	20

Orders:

Order_ID	Order_Name	Item_ID
Geor	George	Veg
Kate	Kate	Chi
Rahu	Rahul	Veg
Vika	Vikas	Mar
Andy	Andy	Chi
Mark	Mark	Mar
Bish	Bish	Van
Bobb	Bobby	Cho
Isha	Ishaan	Cho
Adit	Aditi	Cla

Menu:

Item_ID	Item_Name	Quantity	Unit_Cost
Veg	Veggie Burger	12	5.55
Chi	Chicken Burger	15	7.85
Shr	Shroom Burger	22	6.25
Fri	Fries	26	3.49
Cho	Chocolate Shake	55	4.99
Mar	Margherita Pizza	102	12.99
Van	Vanilla Shake	33	4.99
El	El Cubano	56	7.99
Hot	Hot Dog	120	3.99
Cla	Classic Cheesy Stix	37	5.99

Truck_Menu:

Truck_ID	Item_ID
1	Veg
2	Chi
3	Shr
4	Fri
5	Cho
6	Mar
7	Van
8	El
9	Hot
10	Cla

SQL Queries

Select Queries:

1. Find the name of each item and the quantity in stock for each item that the customer Mark had ordered.

```
select menu.Item_Name, menu.Quantity
from menu
inner join orders
on menu.Item_ID = orders.Item_ID
where orders.Order_Name = "Mark";
```

	Item_Name	Quantity
▶	Margherita Pizza	102

   - Food Truck wants to know what item, or items, that Mark had chosen to purchase. In addition, the Food Truck wants to know how much of each item we have left in stock. This query provides the Item Name and the Quantity of the Item left for the item that Mark wants.

2. Find the total amount of revenue generated on burger orders. Round the final result by 2 decimal places.

```
select round(sum(menu.Unit_Cost),2) as BurgerRevenue
from menu
inner join orders
on menu.Item_ID = orders.Item_ID
where menu.Item_Name like "%Burger";
```

	BurgerRevenue
▶	26.80

   - Food Truck wants to know how much total revenue that the trucks had made on burger orders. This way, they can see how much of a portion of their total revenue is done through burger sales. The query only provides the total revenue generated and rounds the value to 2 decimal places to make sure that the result is in dollars.

3. Find the truck names, color, phone numbers, and full names of the employees who work in Chicago.

```
select employees.First_Name, employees.Last_Name, trucks.Truck_Name, trucks.Truck_Color, trucks.Phone_Number
from employees
inner join truck_employees
on employees.employee_ID = truck_employees.employee_ID
inner join trucks
on trucks.truck_ID = truck_employees.truck_ID
where trucks.city = "Chicago";
```

First_Name	Last_Name	Truck_Name	Truck_Color	Phone_Number
Jacob	Herro	Jade	Red	6308807571
Kareena	Jenner	Karen	Red	6309897238
Rich	Wright	Carter	Blue	6306567433
Arturo	Almeida	Dwayne	Red	6308874221

- Tasty Times wants to know information about the trucks that are in Chicago. They want to know the First and Last names of the employees who work there, as well as the truck's name, color, and phone number that they work for. This query provides this information by using an associative entity to connect them. In our scenario, there isn't any duplicated ID information, but the associative entity allows for this to occur and lets us connect the two tables.

4. Using the information provided in the tables, find the minimum and salaries for each of the positions as well as the respective commision rates.

```
select min(Salary) as MinSalary, max(Salary) as MaxSalary, Commission_Rate as CommissionRate
from employees
group by Manager = "Yes", Cook = "Yes", Cashier = "Yes";
```

MinSalary	MaxSalary	CommissionRate
50,000.00	55,000.00	25%
30,000.00	38,000.00	15%
20,000.00	24,000.00	5%

- Tasty Times wants to know what the minimum and maximum salaries along with the commision rate are for each of the positions. Since the positions were done in a "yes" or "no" layout in the database, we needed to group them based on if they said yes to their respective position. The query provides us with the minimum and maximum salaries for the manager, cook, and cashier positions along with their respective commision rate

5. Find out if there is a truck in Chicago that sells the Veggie Burger, along with the price of the Veggie Burger, the Truck Name, and the Truck Color.

```
select menu.Item_name, menu.Unit_cost, trucks.Truck_Name, trucks.Truck_Color
from menu
inner join truck_menu
on menu.Item_ID = truck_menu.Item_ID
inner join trucks
on trucks.truck_ID = truck_menu.truck_ID
where trucks.city = "Chicago"
and menu.Item_ID = "Veg";
```

Item_name	Unit_cost	Truck_Name	Truck_Color
Veggie Burger	5.55	Jade	Red

- For the Tasty Times Food Service, each food truck only sells one item for now. For this query, we wanted to see if the truck that sold Veggie Burgers was in Chicago, and if there was, we were to retrieve the Item Name, the Unit Cost, the Truck Name, and the Truck color. There was a truck in Chicago that sold the Veggie Burger and, therefore, we received a result from it.

6. Find the name and quantity of each item sold and the name of the truck that sold the item. Sort the items in ascending order.

```
select menu.Item_name as ItemName, count(orders.Item_ID) as QtySold, trucks.Truck_Name as TruckName
from menu
inner join orders
on menu.Item_ID = orders.Item_ID
inner join truck_menu
on orders.Item_ID = truck_menu.Item_ID
inner join trucks
on truck_menu.Truck_ID = trucks.Truck_ID
group by menu.Item_Name
order by count(orders.Item_ID) desc;
```

ItemName	QtySold	TruckName
Veggie Burger	2	Jade
Chicken Burger	2	Simon
Margherita Pizza	2	Kit
Chocolate Shake	2	Carter
Vanilla Shake	1	Corey

Tasty Times Food Service wants to know what they have sold, how much they have sold of each product, in descending order, and what specific truck has sold it. This query provides us with each item that a customer had purchased, the number of times that the customer had purchased the item, and the name of the truck that sold the item or items.

7. Find only the First Name and Last Name of the managers of the company.

```
134 •   Select First_Name,Last_Name,Manager
135     from Employees
136     where Manager ="Yes";
137
```

100% 1:137 1 error found

Result Grid Filter Rows: Search

First_Name	Last_Name	Manager
Jacob	Herro	Yes
Rich	Wright	Yes

- In the employees table, each employee has a different role in the business. For this query, we want to know the first name and last name of the employees who are assigned the role of managers in the company.

8. Find only the First Name and Last Name of the cook of the company.

```
139 •   Select First_Name,Last_Name,Cook
140     from Employees
141     where Cook ="Yes";
```

100% 1:142 1 error found

Result Grid Filter Rows: Search

First_Name	Last_Name	Cook
James	Tomkins	Yes
Kareena	Jenner	Yes
Jonathan	Day	Yes

- In the employees table each employee has a different role. For this query, we wanted to know the first name and last name of the employees who are assigned the role of cooks in the company.

9. Find only the First Name and Last Name of the cashiers of the company.

```
143 •   Select First_Name,Last_Name,Cashier
144     from Employees
145     where Cashier ="Yes";
```

100% 1:146 1 error found

Result Grid Filter Rows: Search

First_Name	Last_Name	Cashier
Kylie	Kapoor	Yes
Mike	Johnson	Yes
Bob	Odenkirk	Yes
Arturo	Almeida	Yes
Maria	Gomez	Yes

- In the employees table each employee has a different role.For this query, we wanted to know the first name and last name of the employees who are assigned the role of Cashier in the company.

10. Find out the total salary given to all the employees by the company.

```
164 •   Select sum(salary) as TotalSalary
165     from employees;
```

TotalSalary
316000.00

- In the employees table we have the column for the salary of different employees. From this query, we can find out the operating expense of the company spent in the form of salaries to the hired employees.

Update, Delete, and Drop Queries:

1. Update the truck with a Truck ID of 1 to zip code of 60611.

```
Update Trucks
set Zipcode = 60611
where Truck_ID = 1;
```

- Here, we want to change the zip code of a truck that has the Truck ID of 1. This may be because we incorrectly typed in the zip code when we originally inserted the data. This query will find the truck with an Truck ID of 1 and change its zip code to 60611.

2. Delete the employees with a salary of $21,000.

```
Delete from employees
Where Salary = 21000.00;
```

- Here, we want to delete the employee with a salary of $21,000 because we do not want that employee in our database anymore. The query deletes this employee.

3. Drop the associative entity between trucks and employees.

```
Drop Table trucks_employees;
```

- Here, we want to drop the associative entity table that combines trucks and employees. This may be because we are going to have only one worker for each truck and don't want multiple workers working at trucks anymore.