2022

Database design and implementation in MySQL Server

MGMT 58200 GROUP 14

Ambati Yamini Praneetha
Amol P Pathak
Chen Yuqiu
Mandeep Singh Rahi
Rachana Ramshetty

1. Background

Kung Fu Tea is a chain of milk tea shop founded in Queens, NY on April 30, 2010. As the largest American bubble tea brand, Kung Fu Tea has placed their stores all around the world. Although not all their stores manage their data well. Our client is Kung Fu Tea located in Chauncey Avenue West Lafayette. They have been operated for 3 years as an outlet of Kung Fu Tea, while they still face supply chain issue and high customer churn rate due to unmet customer needs.

The data table they currently have is as follows:

Sales: Contains the sale volume for each product each month.

The attributes in this entity are

Sales_ID: Primary key that identifies the sales

Product_ID: Primary key that identifies the product

Product_Name: Name of the product

Product_series: Category of each product

ProductPrice: Corresponding price for each product

Sale_Vol: Sale volume for each product

Outlet_ID: Foreign Key identifies the outlet that the record belongs to

Shift: There are two shifts per day which are morning shift and evening shift. Each shift has exactly two employees. This table contains all shifts and their corresponding employees.

The attributes in this entity are

Shift_ID: Primary key that identifies the shifts

Date: The date of each shift

Morning_Night: Whether the shift is a morning shift or night shift

Employee_ID: Foreign key that identifies employee who worked in a specific shift

Employee_Name: Name of the employee who worked in a specific shift

Employee2_ID: Foreign key that identifies employee who worked in a specific shift

Employee2_Name: Name of the employee who worked in a specific shift

Outlet_ID: Foreign Key identifies the outlet that the record belongs to

Employee: Contains all employee information and number of shift they take each month

The attributes in this entity are

Employee_ID: Primary Key identifies employees

Employee_Name: Name of the employee

Hourly_Wage: The hourly salary of the employee

Status: Whether the employee is currently available for work

Outlet_ID: Foreign Key identifies the outlet that the record belongs to

Order: Contains the information about ordering raw material.

The attributes in the entity are

Order_ID: Primary key identifies orders

Material_ID: Foreign key identifies Material

Supplier_Address: Adress where the material shipping from

Contact_number: Phone number of the supplier

Order_quantity: Number of each material in a specific order

Material_Cost: Unit cost of each material in the order

Outlet_ID: Foreign Key identifies the outlet that the record belongs to

Material: Recorded material usage.

The attributes are

Material_ID: Primary key identifies materials

Material_Name: Name of the material

Quantity_Usage : number of units used

Outlet_ID: Foreign Key identifies the outlet that the record belongs to

Outlets: There are thousands of outlets under the brand Kung Fu Tea. It is not the table provided by our client. We only focus on the outlet located in Chauncy Avenue West Lafayette. By creating this table and adding Outlet_ID as a foreign key in all previous tables, we can easily connect all the tables together. If they plan to operate another outlet, they can easily insert the latest
information in the database by updating the information in this table.

The attributes in this entity are

Outlet_ID: Primary Key identifies outlets

Outlet_City: City of the outlet

Outlet_Street: street the outlet locates

Outlet_State: State the outlet locates

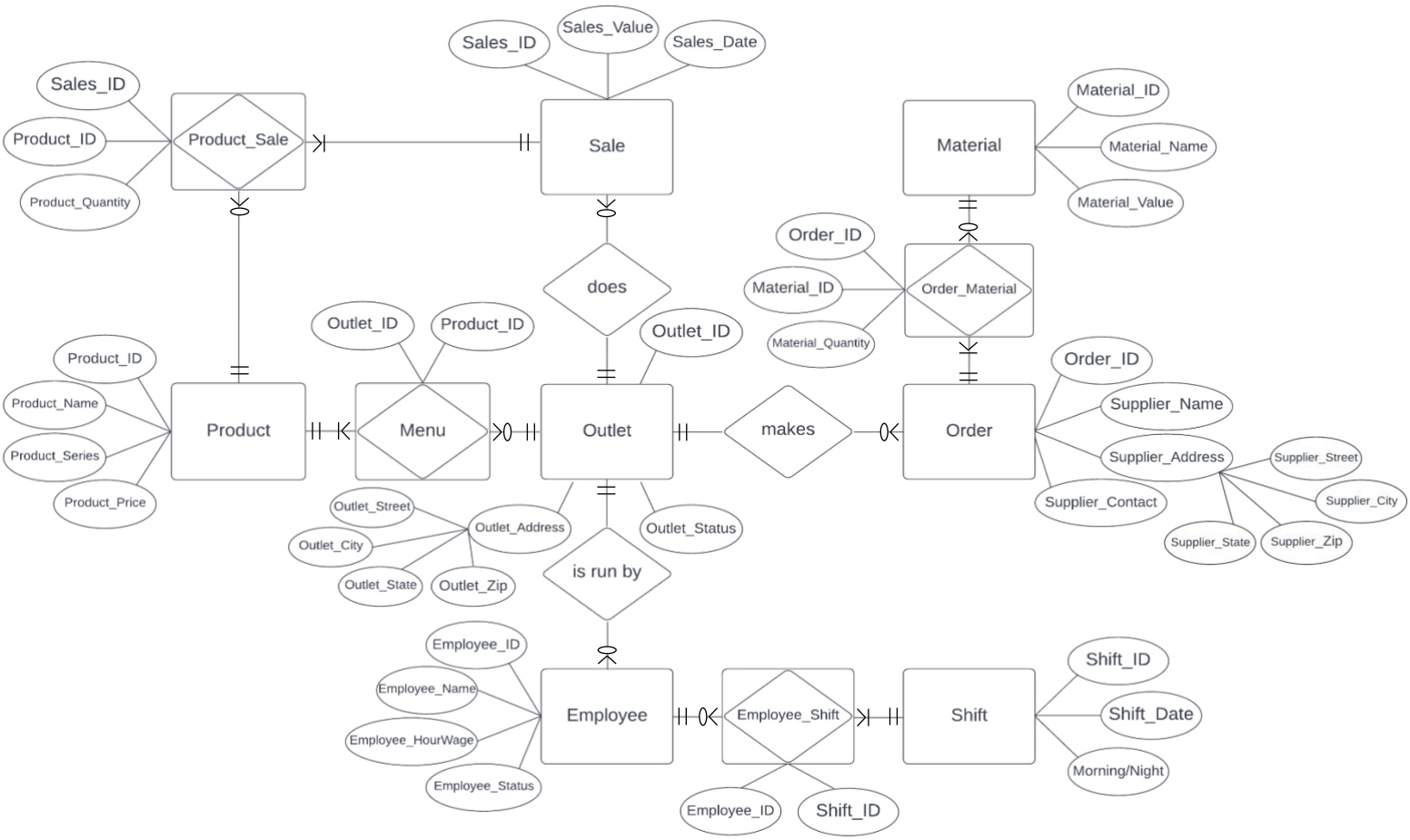
Outlet_Zip: Corresponding zip code

Status: If the store is currently operating by the database owner

2. Introduction

Their various data such as product sale volume, expenses, material usage, are recorded separately in different tables and unorganized. It is difficult for the manager to combine different data for analysis to predict the demand of each product and the order quantity of various materials. It also often exposes them to supply chain issues. Our group want to design a database for them so they can easily combine all kinds of data for analysis.

3. Conceptual Data Modelling: ERD



- Outlets can have zero to many Products, whereas Products must have one to many Outlets.
- Outlets does zero to many Sales, whereas Sale belong to only one Outlet.
- Products can have zero to many Sales, whereas Sale can have one to many Products.
- Outlets are run by zero to many Employees, whereas Employees belong to only one Outlet.
- Employees can have zero to many Shifts, whereas Shifts can have one or many Employees.
- Outlets can have zero to many Orders, whereas Order belong to only one Outlet.
- Orders can have one to many Materials, whereas Materials can belong to zero to many Orders.

4. Relational Data Model: Relational Schema

Outlet(Outlet_ID, Outlet_Status, Outlet_Street, Outlet_City, Outlet_State, Outlet_Zip)

Order(Order_ID, Supplier_Contact, Supplier_Name, Supplier_Street, Supplier_City, Supplier_State, Supplier_Zip, Outlet_ID)

Material(Material_ID, Material_Name, Material_Value)

Employee(Employee_ID, Employee_Name, Employee_HourWage, Employee_Status, Outlet_ID)

Shift(Shift_ID, Shift_Date, Morning_Night)

Product(Product_ID, Product_Name, Product_Series, Product_Price)

Sale(Sales_ID, Sales_Date, Sales_Value, Outlet_ID)

Associative Tables:

Order_Material(Order_ID, Material_ID, Material_Quantity)

Shift_Employee(Shift_ID, Employee_ID)

Menu(Outlet_ID, Product_ID)

Product_Sale(Sales_ID, Product_ID, Product_Quantity)

5. Normalization Process

No Normal Form: None of the tables are 0NF, there is not any multi-valued and composite attribute

First Normal Form: All tables are in 1NF, each tuple contains exactly one value for each attribute.

Second Normal Form:

> **Functional Dependency:** For the sales table, there are two primary keys which are Sales_ID and Product_ID. The Sale_Vol depends on both Sales_ID and Product_ID, the Product_Name, SalePrice and Product_Series only depends on Product_ID.
>
> Therefore, The Sales table is not in 2NF because not all non-key attributes are fully functionally dependent on the entire primary key.

Decomposition to 2NF: To normalize the Sales table, we divide it into two tables: A Sale table contains only Sales_ID and Sale_Vol and Product table contains the rest of information.

Third Normal Form: All tables are in 3NF, there is no transitive dependency for non-key attributes.

6. SQL Implementation

The following are a few of the queries we have implemented based on the business requirement:

1. Top Products sold (Top 5 products sold):
   ```
   select p.*,sum(sp.Product_Quantity) as Total_Sale from product p
   left join sales_product sp on sp.product_ID = p.product_ID
   group by p.product_ID
   order by sum(sp.Product_Quantity) desc
   Limit 5;
   ```

Product_ID	Product_name	Product_seri...	Price	Total_Sale
107	KF Milk Tea	MilkTea	5.75	5755
109	Coffee Milk Tea	MilkTea	5.95	4470
111	Almond Milk Tea	MilkTea	5.95	2984
110	Taro Milk Tea	MilkTea	5.95	2594
108	KF Milk Green/Oolong Tea	MilkTea	5.75	2463

2. Most ordered material (Top 5)
   ```
   select m.*,sum(om.quantity) as Total_Quantity from material m
   left join order_material om on om.material_ID = m.material_ID
   group by m.material_ID
   order by sum(om.quantity) desc
   Limit 5;
   ```

Material_ID	Material_Name	Value	Total_Quanti...
KTW-I0069	Cane Sugar	255	100
KTW-I0065	Popping Bubble - Coffee (4 bottles / case)	138	69
KTW-I0029	Winter Melon Syrup (4 bottless / case)	80	66
KTW-I0025	Strawberry Syrup (4 bottless / case)	400	63
KTW-S0121-1	PET Plastic Cold Cup 24 oz (*NEW*) (1000 PC...	1596	62

3. Cost of Materials ordered over the entire time period
   ```
   select m.*,sum(om.quantity) as Total_Quantity, m.value*sum(om.quantity) as Total_Amount from material m
   left join order_material om on om.material_ID = m.material_ID
   group by m.material_ID
   order by Total_Amount desc;
   ```

Material_ID	Material_Name	Value	Total_Quanti...	Total_Amount
KTW-I0053	Milk Powder (10 bags / case) (Non-Dairy)	1920	57	109440
KTW-S0121-1	PET Plastic Cold Cup 24 oz (*NEW*) (1000 PC...	1596	62	98952
KTW-S0124-1	PET Plastic Cold Cup 16 oz (*NEW*) (1000 PC...	904	37	33448
KTW-I0076	Bubble - Black (6 bags / case)	510	52	26520
KTW-I0066	Popping Bubble - Mango (4 bottles / case)	552	47	25944
KTW-I0069	Cane Sugar	255	100	25500
KTW-I0025	Strawberry Syrup (4 bottless / case)	400	63	25200
KTW-I0057	Taro Syrup (15 bottles / case)	560	42	23520
KTW-I0008	ESSSE Coffee Bean (6 bags / case)	384	50	19200
KTW-I0030	Mango Coconut Jelly (4 bottless / case)	366	52	19032
KTW-I0078	Crystal Ball - Lychee (6 Bags/CTN)	335	51	17085
KTW-I0031	Mango Juice w/ Fruit (4 bottles / case)	320	48	15360
KTW-I0094	Hershey's Chocolate Powder - 910g/ 20bag/ CTN	266	56	14896
KTW-I0001	Black Tea (40 bags / case)	446	32	14272
KTW-I0081	Blueberry & Rosehip Powder (20 Bags/ Case)	196	56	10976
KTW-S0030	Take out bag (2-cup) (10 packs/case)	246	41	10086
KTW-I0051	Lychee Jam (6 bottles / case)	176	56	9856
KTW-I0080	Pineapple B (New)(15 Bottles/Case)	394	25	9850
KTW-I0065	Popping Bubble - Coffee (4 bottles / case)	138	69	9522
KTW-S0027	Straw 12 X 21 (2250 pcs/case)	165	51	8415
KTW-I0038	Brown Sugar Syrup (4 bottles / case)	146	51	7446
KTW-I0029	Winter Melon Syrup (4 bottless / case)	80	66	5280
KTW-I0109	Kiwi Jam (12 bottles/case)	153	32	4896
KTW-I0110	Guava Syrup (12 bottles/case)	98	46	4508
KTW-I0047	Coconut Powder (10 bags / case)	107	39	4173
KTW-I0055-1	Peach Jam w. Pulp (12 bottles / case) **NEW**	104	39	4056
KTW-I0013	Herbal Juice (6 bottles / case)	90	35	3150
KTW-S0101	Cup Sticker (Revel) (5 rolls / pack)**	48	44	2112
KTW-H0008	Hot Pot Strainer	10	51	510

4. Total sales value grouped by Product_ID
   select p.*,sum(sp.Product_Quantity) as Total_Sale from product p
   left join sales_product sp on sp.product_ID = p.product_ID
   group by p.product_ID
   order by p.product_ID;

Product_ID	Product_name	Product_seri...	Price	Total_Sale
100	KF Black Tea	CLASSIC	4.85	173
101	KF Green/Oolong Tea	CLASSIC	4.85	115
102	KF Pomelo Tea	CLASSIC	4.85	105
103	Winter Melon Tea	CLASSIC	5	181
104	Winter Melon Green Tea	CLASSIC	5	258
105	KF Honey Tea	CLASSIC	4.85	93
106	Logan Jujube Tea	CLASSIC	5	76
107	KF Milk Tea	MilkTea	5.75	5755
108	KF Milk Green/Oolong Tea	MilkTea	5.75	2463
109	Coffee Milk Tea	MilkTea	5.95	4470
110	Taro Milk Tea	MilkTea	5.95	2594
111	Almond Milk Tea	MilkTea	5.95	2984
112	Coconut Milk Tea	MilkTea	5.95	1436
113	Oreo Milk Tea	MilkTea	5.95	1913
114	Wintermelon Milk Green...	MilkTea	5.95	991
115	Signature Coffee	COFFEE	5.5	1768
116	Italian Ice Mocha	COFFEE	5.5	965
117	Ice Cappuccino	COFFEE	5.5	567
118	Ice Coffee Latte	COFFEE	5.5	494
119	Ice Caramel Macchiato	COFFEE	5.5	460
120	Lemon Black/Green Tea	PUNCH	5	115
121	Honey Lemonade	PUNCH	5	213
122	Orange Green Tea	PUNCH	5.5	175
123	Grapfruit Green Tea	PUNCH	5.5	450
124	PassionFruit Green Tea	PUNCH	5.5	790
125	Sunshine Pineapple Tea	PUNCH	5.5	1083
126	Mango Grenn Tea	PUNCH	5.5	1894
127	Lychee Tea	PUNCH	5.5	1178
128	Peach Oolong Tea	PUNCH	5.5	1164
129	Strawberry Lemon Green...	PUNCH	5.75	1833
130	Taro Slush	SLUSH	6.25	959
131	Mango Slush	SLUSH	6.25	1356
132	Passionfruit Slush	SLUSH	6.25	397
133	Pineapple Slush	SLUSH	6.25	159
134	Strawberry Milk Slush	SLUSH	6.25	418
135	Macha Red Bean Slush	SLUSH	6.25	100
136	Signiture Coffee Slush	SLUSH	6.25	93
137	Italian Mocha Slush	SLUSH	6.25	282
138	Caramel Macchiato Slush	SLUSH	6.25	365
139	Pina Colada Slush	SLUSH	6.25	293
140	Chai Milk	MILK	5.75	247
141	Ginger Milk	MILK	5.75	153
142	Matcha Milk	MILK	5.75	117
143	Wintermelon Milk	MILK	5.75	53
144	Yogurt Green Tea	YOGURT	6.25	59
145	Yogurt Orange Juice	YOGURT	6.25	90
146	Yogurt Grapefruit Juice	YOGURT	6.25	100
147	Yogurt Lamonade	YOGURT	6.25	52

5. Employee with max shifts
   select e.employee_ID, e.employee_name, count(*) as Shift_Count from employee e
   left join shift_employee se on e.employee_ID = se.employee_ID
   group by e.employee_ID
   order by count(*) desc
   limit 1;

employee_ID	employee_name	Shift_Count
1001	Hong C	213

6. Are there any employees with no shift
```
select e.employee_ID, e.employee_name from employee e
left join shift_employee se on e.employee_ID = se.employee_ID
group by e.employee_ID
Having count(se.shift_ID) =0
order by e.employee_ID;
```

employee_ID	employee_name

7. Products not sold at all
```
select p.*,sum(sp.Product_Quantity) as Total_Sale from product p
left join sales_product sp on sp.product_ID = p.product_ID
group by p.product_ID
Having sum(sp.Product_Quantity) =0;
```

Product_ID	Product_name	Product_seri...	Price	Total_Sale

8. No. of sales with sales value greater than $500 in a day
```
select count(sales_ID) from sales
where sales_value > 500;
```

count(sales_ID)
87

9. No. of products sold starting with “coffee” in product_name
```
select p.*,sum(sp.Product_Quantity) as Total_Sale from product p
left join sales_product sp on sp.product_ID = p.product_ID
where product_name like 'coffee%'
group by p.product_ID
order by sum(sp.Product_Quantity) desc;
```

Product_ID	Product_name	Product_seri...	Price	Total_Sale
109	Coffee Milk Tea	MilkTea	5.95	4470

10. Total sale volume
    select sum(sales_value) as Total_Sale from sales;

Total_Sale
243204.2000000001

11. Sum of sales value grouped by month
    select month(sales_Date) as Sale_Month,sum(sales_Value) as Total_sale_Value
    from sales
    group by month(sales_Date)
    order by sum(sales_Value) desc;

Sale_Month	Total_sale_Value
4	37153.25
8	34497.1
5	31344.35
2	30738.65
1	29696.95
3	29192.8
7	27052.9
6	23528.2

12. Products sold and their quantities when there is maximum sale
    select p.*, s.*, sp.product_quantity from product p
    left join sales_product sp on sp.product_ID = p.product_ID
    left join (select * from sales order by sales_value desc Limit 1) s on s.sales_ID = sp.sales_ID
    where s.sales_ID is not null;

Product_ID	Product_name	Product_seri...	Price	Sales_ID	Sales_Value	sales_Date	outlet_ID	product_quant...
107	KF Milk Tea	MilkTea	5.75	137	9011.1	2022-04-09 00:00:00	1	841
111	Almond Milk Tea	MilkTea	5.95	137	9011.1	2022-04-09 00:00:00	1	583
140	Chai Milk	MILK	5.75	137	9011.1	2022-04-09 00:00:00	1	62
134	Strawberry Milk Slush	SLUSH	6.25	137	9011.1	2022-04-09 00:00:00	1	56

13. Maximum average value of material across all the orders
    select avg(m.value) from orders o
    left join order_material om on o.order_ID = om.order_ID
    left join material m on m.material_ID = om.material_ID
    group by o.order_ID
    order by avg(m.value) desc
    limit 1;

avg(m.value)
1920.0000

14. Most ordered product within each series

```
With top_sold as
(select p.*,sp.Total_Quantity, Row_Number() over(partition by p.product_series order by sp.Total_Quantity desc) as Max_Quantity
from product p
left join (select product_ID,sum(product_quantity) as Total_Quantity from sales_product group by product_ID) sp
on p.product_ID = sp.product_ID)
select product_ID, Product_Name, Price, Total_Quantity from top_sold where Max_Quantity = 1;
```

product_ID	Product_Name	Price	Total_Quantity
104	Winter Melon Green Tea	5	258
115	Signature Coffee	5.5	1768
140	Chai Milk	5.75	247
107	KF Milk Tea	5.75	5755
126	Mango Grenn Tea	5.5	1894
131	Mango Slush	6.25	1356
146	Yogurt Grapefruit Juice	6.25	100

7. Commentary

Our clients, Kung Fu Tea, were struggling with unorganized data. Our team understood their business scenario and designed a database solution that solves their business problem and is also future proof.

In order to do so, we prepared an ER Diagram and Relational Schema that represents their business scenario. We also ensured that the data is normalized to avoid any data redundancy. While designing the database, we added a new entity for Outlets to incorporate the possibility of future expansion of the company. The team organized the data and implemented the database on MySQL Server. Later, we executed several queries on the database as testing.

Our solution will improve the organization of data for our clients by providing normalized database structure. With the implementation of database on MySQL Server, in future, our clients could also integrate additional applications to easily interact with this data that could solve business challenges such as demand planning, sales analysis, employee information tracking, etc.