

LVX
VERITAS
VIRTUS

Chinese Restaurant Management System Database

Team 9: Fang-Ying Tien, Jingzhi Zhang,
Yaqing Peng, Yiyun Lyu, Yi-Chieh Huang



Agenda



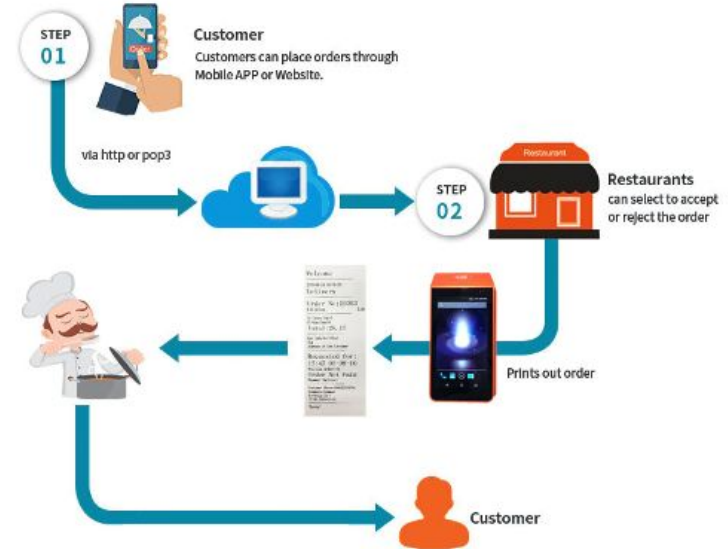
- 1.Fang-Ying Tien: Design Document
- 2.Yiyun Lyu: ERD
- 3.Jingzhi Zhang: SQL DDL
- 4.Yaqing Peng: Views
- 5.Yi-Chieh Huang: Visualization

Purpose :

- Systematically organize and preserve critical restaurant data
- Facilitate efficient management of dish offerings, customer profiles, customer orders, and shopping carts
- Enhance customer experience by streamline the process of browsing menus, placing orders and making payment in the restaurant

Business Problem Addressed :

- Streamline order management and menu organization
- Generate reports to analyze the overall business performance



13 Entities:

- Category: Organize menu by categorizing dishes
- Dish: Manage dish offerings with unique IDs, names, prices, and statuses
- DishSetMeal: Establish connections between dishes and set meals
- SetMeal: Define bundled meal options
- Payment: Document customer payment details
- Flavor: Represent customer preferences for dish flavors
- Discount: Manage promotional offers on dishes
- OrderDetail: Detail each order and link to shopping cart items
- Order: Store information about customer orders
- Customer: Store customer profiles and account information
- Address: Store customer billing addresses
- Billing: Capture information related to bills paid by customers
- ShoppingCartItem: Enable customers to add dishes and set meals to shopping carts

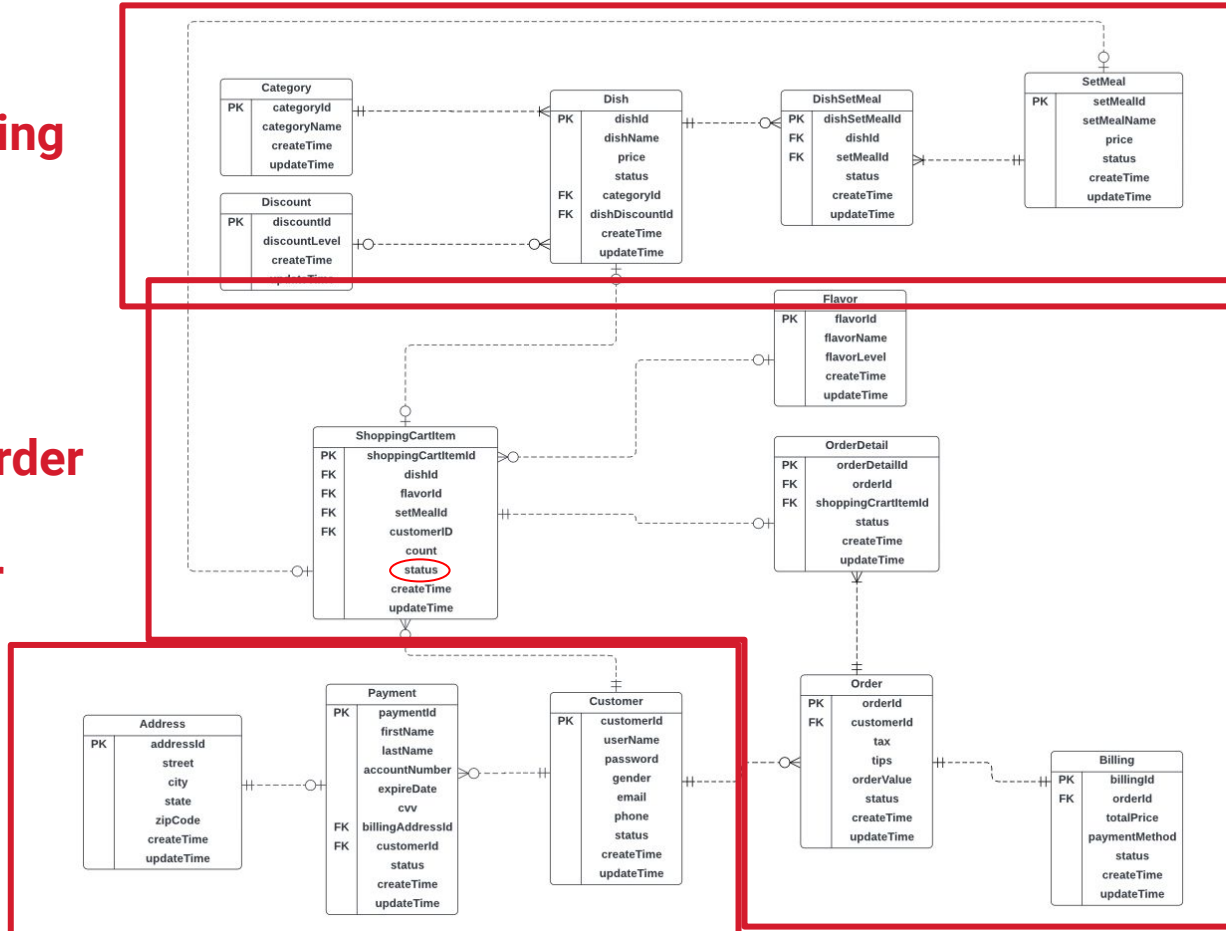
Agenda

- 1.Fang-Ying Tien: Design Document
- 2.Yiyun Lyu: ERD
- 3.Jingzhi Zhang: DDL
- 4.Yaqing Peng: Views
- 5.Yi-Chieh Huang: Visualization

Dish Offering

Order

Customer



Agenda

- 1.Fangying Tien: Design Document
- 2.Yiyun Lyu: ERD
- 3.Jingzhi Zhang: DDL
- 4.Yaqing Peng: Views
- 5.Yi-Chieh Huang: Visualization

Table Creation :

PK, default time, default status and when status change

13 tables including Category, Discount, Dish, SetMeal, DishSetMeal, Flavor, Address, Customer, Payment, ShoppingCartItem, Order, OrderDetail and Billing

'ON' → 'OFF'

```
CREATE TABLE dbo.Dish (
    dishId INT IDENTITY NOT NULL PRIMARY KEY,
    dishName VARCHAR(255) NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    status VARCHAR(50) NOT NULL DEFAULT 'ON',
    categoryId INT NOT NULL REFERENCES Category(categoryId),
    dishDiscountId INT REFERENCES Discount(discountId),
    createTime DATETIME DEFAULT GETDATE(),
    updateTime DATETIME DEFAULT GETDATE()
);
```

```
CREATE TABLE SetMeal (
    setMealId INT IDENTITY NOT NULL PRIMARY KEY,
    setMealName VARCHAR(255) NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    status VARCHAR(50) NOT NULL DEFAULT 'ON',
    createTime DATETIME DEFAULT GETDATE(),
    updateTime DATETIME DEFAULT GETDATE()
);
```

```
CREATE TABLE DishSetMeal (
    dishSetMealId INT IDENTITY NOT NULL PRIMARY KEY,
    dishId INT NOT NULL REFERENCES Dish(dishId),
    setMealId INT NOT NULL REFERENCES SetMeal(setMealId),
    status VARCHAR(50) NOT NULL DEFAULT 'ON',
    createTime DATETIME DEFAULT GETDATE(),
    updateTime DATETIME DEFAULT GETDATE()
);
```

```
CREATE TABLE dbo.Customer (
    customerId INT IDENTITY NOT NULL PRIMARY KEY,
    userName VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL,
    gender VARCHAR(50),
    email VARCHAR(50),
    phone VARCHAR(50),
    status VARCHAR(50) DEFAULT 'ON',
    createTime DATETIME DEFAULT GETDATE(),
    updateTime DATETIME DEFAULT GETDATE()
);
```

```
CREATE TABLE dbo.Payment (
    paymentId INT IDENTITY NOT NULL PRIMARY KEY,
    firstName VARCHAR(255) NOT NULL,
    lastName VARCHAR(255) NOT NULL,
    accountNumber VARCHAR(50) NOT NULL,
    expireDate DATETIME NOT NULL,
    cvv VARCHAR(50) NOT NULL,
    billingAddressId INT NOT NULL
    REFERENCES Address(addressId),
    customerId INT NOT NULL
    REFERENCES Customer(customerId),
    status VARCHAR(50) DEFAULT 'ON',
    createTime DATETIME DEFAULT GETDATE(),
    updateTime DATETIME DEFAULT GETDATE()
);
```

```
CREATE TABLE ShoppingCartItem (
    shoppingCartItemId INT IDENTITY NOT NULL PRIMARY KEY,
    dishId INT REFERENCES Dish(dishId),
    flavorId INT REFERENCES Flavor(flavorId),
    setMealId INT REFERENCES SetMeal(setMealId),
    customerId INT NOT NULL REFERENCES Customer(customerId),
    [count] INT NOT NULL,
    status VARCHAR(50) NOT NULL DEFAULT 'ON',
    createTime DATETIME DEFAULT GETDATE(),
    updateTime DATETIME DEFAULT GETDATE()
);
```

'PROCESSING' → 'COMPLETED'

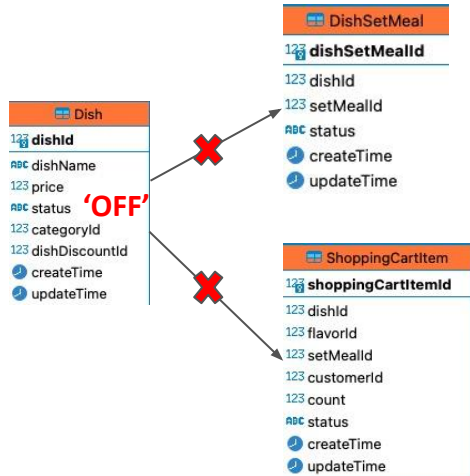
```
CREATE TABLE "Order" (
    orderId INT IDENTITY NOT NULL PRIMARY KEY,
    customerId INT NOT NULL REFERENCES "Customer"(customerId),
    tips AS (orderId * 0.18) PERSISTED,
    orderValue DECIMAL(10, 2),
    tax AS (orderId * 0.1) PERSISTED,
    status VARCHAR(50) DEFAULT 'PROCESSING',
    createTime DATETIME DEFAULT GETDATE(),
    updateTime DATETIME DEFAULT GETDATE()
);
```

```
CREATE TABLE OrderDetail (
    orderDetailId INT IDENTITY NOT NULL PRIMARY KEY,
    shoppingCartItemId INT NOT NULL REFERENCES "shoppingCartItem"(shoppingCartItemId),
    orderId INT NOT NULL REFERENCES "Order"(orderId),
    status VARCHAR(50) DEFAULT 'PROCESSING',
    createTime DATETIME DEFAULT GETDATE(),
    updateTime DATETIME DEFAULT GETDATE()
);
```

```
CREATE TABLE dbo.Billing (
    billingId INT IDENTITY NOT NULL PRIMARY KEY,
    orderId INT NOT NULL
    REFERENCES [Order](orderId),
    paymentMethod VARCHAR(50) NOT NULL,
    status VARCHAR(50) DEFAULT 'PROCESSING',
    createTime DATETIME DEFAULT GETDATE(),
    updateTime DATETIME DEFAULT GETDATE()
);
```




Constraints & Functions



```
CREATE FUNCTION dbo.IsDishAvailable(@dishId INT)
RETURNS VARCHAR(50)
AS
BEGIN
    DECLARE @status VARCHAR(50)

    SELECT @status = status FROM Dish WHERE DishId = @dishId

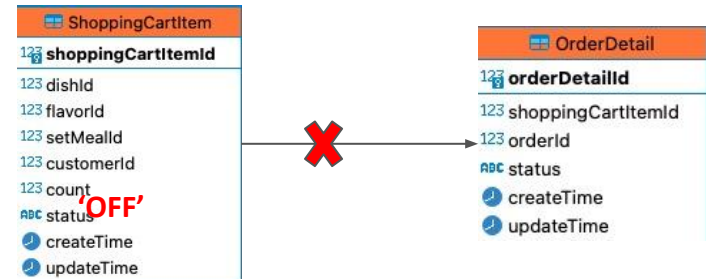
    RETURN @status;
END;
```

—Add a constraint on ShoppingCartItem to prevent inserting the dish that is out of stock

```
ALTER TABLE dbo.[ShoppingCartItem]
ADD CONSTRAINT CHK_Dish_status CHECK (dbo.IsDishAvailable(dishId) = 'ON');
```

—Add a constraint on DishSetMeal to prevent inserting the dish that is out of stock

```
ALTER TABLE dbo.[DishSetMeal]
ADD CONSTRAINT CHK_DishSetMeal_status CHECK (dbo.IsDishAvailable(dishId) = 'ON');
```



```
--- Constraint2: Check The Availability Of Shopping Cart Items
CREATE FUNCTION dbo.fnCheckShoppingCartItemResult(@shoppingCartItemID INT)
RETURNS VARCHAR(10)
AS
BEGIN
    DECLARE @status VARCHAR(50)
    DECLARE @counts INT
    DECLARE @result VARCHAR(10) = 'ON'

    --- Retrieve status and count
    SELECT @status = status, @counts = [count]
    FROM ShoppingCartItem
    WHERE shoppingCartItemID = @shoppingCartItemID;

    IF @status = 'OFF' OR @counts = 0
    BEGIN
        SET @result = 'OFF'
    END
    RETURN @result
END;
```

```
ALTER TABLE OrderDetail ADD CONSTRAINT CheckOrderDetail_ShoppingCartItem CHECK (dbo.fnCheckShoppingCartItemResult(shoppingCartItemID) = 'ON');
```



Triggers & Computed Columns

Shopping Cart item Status

ON → OFF



ShoppingCart

Order

--Trigger1: To switch status in shoppingCartItem to off after adding into orderDetail
DROP TRIGGER ChangeStatusWhenAddToOrder;

```
CREATE TRIGGER ChangeStatusWhenAddToOrder  
ON OrderDetail  
AFTER INSERT  
AS  
BEGIN  
    SET NOCOUNT ON;  
    UPDATE ShoppingCartItem  
    SET status = 'OFF'  
    FROM ShoppingCartItem sci  
    INNER JOIN OrderDetail od  
    ON sci.shoppingCartItemid = od.shoppingCartItemid  
    WHERE sci.status = 'ON';  
END;
```

| | 143 billingId | 123 orderId | 100 paymentMethod | 100 status | 100 createTime | 100 updateTime | 123 totalPrice |
|----|---------------|-------------|-------------------|------------|-------------------------|-------------------------|----------------|
| 1 | 1 | 1 | Credit Card | COMPLETED | 2023-01-01 12:00:00.000 | 2023-01-01 12:00:00.000 | 66.53 |
| 2 | 2 | 2 | Credit Card | COMPLETED | 2023-01-16 12:00:00.000 | 2023-01-16 12:00:00.000 | 25.59 |
| 3 | 3 | 3 | Credit Card | COMPLETED | 2023-01-31 12:50:00.000 | 2023-01-31 12:50:00.000 | 31.99 |
| 4 | 4 | 4 | Credit Card | COMPLETED | 2023-01-31 14:30:00.000 | 2023-01-31 14:30:00.000 | 25.59 |
| 5 | 5 | 5 | Credit Card | COMPLETED | 2023-02-14 08:00:00.000 | 2023-02-14 12:00:00.000 | 31.99 |
| 6 | 6 | 6 | Credit Card | COMPLETED | 2023-02-24 08:00:00.000 | 2023-02-24 12:00:00.000 | 33.27 |
| 7 | 7 | 7 | Credit Card | COMPLETED | 2023-02-28 16:00:00.000 | 2023-02-28 20:00:00.000 | 31.99 |
| 8 | 8 | 8 | Credit Card | COMPLETED | 2023-03-08 08:00:00.000 | 2023-03-08 12:00:00.000 | 25.59 |
| 9 | 9 | 9 | Credit Card | COMPLETED | 2023-03-10 08:00:00.000 | 2023-03-10 08:00:00.000 | 701.82 |
| 10 | 10 | 10 | Credit Card | COMPLETED | 2023-04-08 08:00:00.000 | 2023-04-08 12:00:00.000 | 95.96 |

```
-- Computed Column (totalPrice in Table Billing)  
CREATE FUNCTION dbo.CalculateTotalPrice (@orderId int)  
RETURNS DECIMAL(10, 2)  
AS  
BEGIN  
    DECLARE @totalPrice DECIMAL(10, 2);  
  
    SELECT @totalPrice = SUM(tax + tips + orderValue)  
    FROM dbo.[Order]  
    WHERE orderId = @orderId;  
  
    RETURN ISNULL(@totalPrice, 0);  
END;  
  
ALTER TABLE dbo.Billing  
ADD totalPrice AS dbo.CalculateTotalPrice(orderId);
```

Data Insertion

-- Insert data into Category table

```
INSERT INTO dbo.Category (categoryName, createTime, updateTime)
VALUES
```

```
('Appetizer', '2022-01-01', '2022-01-01'),
('Desert', '2022-01-01', '2022-01-01'),
('Soup', '2022-01-01', '2022-01-01'),
('Main', '2022-01-01', '2022-01-01'),
('Drink', '2022-01-01', '2022-01-01'),
('Side Dish', '2022-01-01', '2022-01-01'),
('Breakfast', '2022-01-01', '2022-01-01'),
('Vegan', '2022-01-01', '2022-01-01'),
('BBQ', '2022-01-01', '2022-01-01'),
('Hot Pot', '2022-01-01', '2022-01-01');
```

-- Insert data into Dish table

VALUES

```
(0.90, '2022-01-01', '2022-01-01'),
(0.80, '2022-01-01', '2022-01-01'),
(0.89, '2022-01-01', '2022-01-01'),
(0.88, '2022-01-01', '2022-01-01'),
(0.87, '2022-01-01', '2022-01-01'),
(0.86, '2022-01-01', '2022-01-01'),
(0.85, '2022-01-01', '2022-01-01'),
(0.84, '2022-01-01', '2022-01-01'),
(0.83, '2022-01-01', '2022-01-01'),
(0.82, '2022-01-01', '2022-01-01'),
(0.81, '2022-01-01', '2022-01-01');
```

--Insert data into Dish table(with discount)

```
INSERT INTO dbo.Dish (dishName, price, status, categoryId, dishDiscountId, createTime, updateTime)
VALUES
```

```
('Kung Pao Chicken', 10.00, 'ON', 4, 1, '2022-01-01', '2022-01-01'),
('Mapo Tofu', 15.00, 'ON', 4, 2, '2022-01-01', '2022-01-01'),
('Wonton Soup', 20.00, 'ON', 3, 1, '2022-01-01', '2022-01-01'),
('Braised Pork', 30.00, 'ON', 4, 1, '2022-01-01', '2022-01-01'),
('Spring Rolls', 25.00, 'OFF', 2, 2, '2022-01-01', '2022-01-01'),
('Sweet and Sour Pork', 30.00, 'ON', 4, 1, '2022-01-01', '2022-01-01'),
('Hot and Sour Soup', 20.00, 'ON', 3, 1, '2022-01-01', '2022-01-01'),
('Sichuan Noodles', 15.00, 'ON', 4, 2, '2022-01-01', '2022-01-01'),
('Peking Duck Rolls', 25.00, 'ON', 1, 1, '2022-01-01', '2022-01-01'),
('Stir-Fried Green Beans', 12.00, 'ON', 4, 2, '2022-01-01', '2022-01-01'),
('Spring Onion Pancake', 8.00, 'ON', 1, 2, '2022-01-01', '2022-01-01'),
('Vegetable Spring Rolls', 10.00, 'OFF', 1, 1, '2022-01-01', '2022-01-01'),
('Crispy Shrimp Wontons', 12.00, 'ON', 1, 1, '2022-01-01', '2022-01-01'),
('Cashew Chicken', 26.00, 'ON', 4, 2, '2022-01-01', '2022-01-01'),
('Eggplant with Garlic Sauce', 24.00, 'ON', 4, 1, '2022-01-01', '2022-01-01'),
('Hot and Spicy Beef', 30.00, 'ON', 4, 2, '2022-01-01', '2022-01-01'),
('Sweet and Sour Fish', 27.00, 'ON', 4, 1, '2022-01-01', '2022-01-01'),
('Cantonese Chow Mein', 25.00, 'ON', 4, 2, '2022-01-01', '2022-01-01'),
('Stir-Fried Tofu with Vegetables', 23.00, 'ON', 4, 1, '2022-01-01', '2022-01-01'),
('Hunan Pork', 31.00, 'ON', 4, 2, '2022-01-01', '2022-01-01'),
('Shrimp with Lobster Sauce', 25.00, 'ON', 4, 1, '2022-01-01', '2022-01-01'),
('Kung Pao Shrimp', 22.00, 'ON', 4, 1, '2022-01-01', '2022-01-01'),
('General Tso's Chicken', 20.00, 'ON', 4, 1, '2022-01-01', '2022-01-01');
```

INSERT INTO SetMeal (setMealName, price, status, createTime, updateTime) VALUES

```
('Special A', 25.99, 'ON', '2022-01-01T08:00:00', '2022-01-01T12:00:00'),
('Farewell A', 24.99, 'ON', '2022-02-09T08:00:00', '2022-02-09T12:00:00'),
('Farewell B', 24.99, 'ON', '2022-02-09T08:00:00', '2022-02-09T12:00:00'),
('Farewell C', 24.99, 'ON', '2022-03-28T08:00:00', '2022-03-28T12:00:00'),
('Fresh A', 24.99, 'ON', '2022-04-01T08:00:00', '2022-04-01T12:00:00');
```

| | Table Name | Dish | Discount | Category | Setmeal | Dish Setmeal | Shopping Cart | Flavor | Order Detail | Order | Billing | Customer | Payment | Address |
|--|----------------|-------|----------|----------|---------|--------------|---------------|--------|--------------|-------|---------|----------|---------|---------|
| | (ON/) Total | 38/40 | 11 | 10 | 13/15 | 18/18 | 0/325 | 10 | 0/325 | 0/105 | 0/105 | 10/15 | 10/15 | 15 |

-- Insert active Customer data

```
INSERT INTO dbo.Customer (userName, password, gender, email, phone, createTime, updateTime)
VALUES
```

```
('john_doe', 'password123', 'Male', 'john.doe@example.com', '123-456-7890', '2022-01-01', '2022-01-01'),
('jane_smith', 'securePW456', 'Female', 'jane.smith@example.com', '987-654-3210', '2022-01-01', '2022-01-01'),
('mike_jones', 'password', 'Male', 'mike.jones@example.com', '555-123-4567', '2022-01-01', '2022-01-01'),
('sarah_green', 'green123', 'Female', 'sarah.green@example.com', '333-999-8888', '2022-01-01', '2022-01-01'),
('alex_carter', 'carter456', 'Male', 'alex.carter@example.com', '777-222-1111', '2022-01-01', '2022-01-01'),
('lisa_brown', 'brownPass', 'Female', 'lisa.brown@example.com', '444-777-0000', '2022-01-01', '2022-01-01'),
('tom_wilson', 'wilson789', 'Male', 'tom.wilson@example.com', '666-444-2222', '2022-01-01', '2022-01-01'),
('emily_adams', 'adams2020', 'Female', 'emily.adams@example.com', '111-888-9999', '2022-01-01', '2022-01-01'),
('chris_evans', 'evansPWD', 'Male', 'chris.evans@example.com', '999-000-3333', '2022-01-01', '2022-01-01'),
('amy_white', 'whitePass', 'Female', 'amy.white@example.com', '222-555-6666', '2022-01-01', '2022-01-01');
```

INSERT INTO Flavor (flavorName, flavorLevel, createTime, updateTime) VALUES

```
('Mild', 1, '2022-01-01', '2022-01-01'),
('Medium', 2, '2022-01-01', '2022-01-01'),
('Spicy', 3, '2022-01-01', '2022-01-01'),
('Extra Spicy', 4, '2022-01-01', '2022-01-01'),
('Normal Sugar', 3, '2022-01-01', '2022-01-01'),
('Less Sugar', 2, '2022-01-01', '2022-01-01'),
('No Sugar', 1, '2022-01-01', '2022-01-01'),
('Normal Ice', 3, '2022-01-01', '2022-01-01'),
('Light Ice', 2, '2022-01-01', '2022-01-01'),
('No Ice', 1, '2022-01-01', '2022-01-01');
```

Agenda

- 1.Fangying Tien: Design Document
- 2.Yiyun Lyu: ERD
- 3.Jingzhi Zhang: DDL
- 4.Yaqing Peng: Views
- 5.Yi-Chieh Huang: Visualization

4.View1: Monthly Turnover of 2023

```
CREATE VIEW MonthlyTurnover AS
SELECT MONTH(CreateTime) AS Month, SUM(TotalPrice) AS MonthlyRevenue
FROM dbo.Billing
WHERE Status = 'COMPLETED'
GROUP BY MONTH(CreateTime);
```

| 123 Month | 123 MonthlyRevenue |
|-----------|--------------------|
| 1 | 1,873.5 |
| 2 | 1,511.86 |
| 3 | 1,662.86 |
| 4 | 584.72 |
| 5 | 2,095.31 |
| 6 | 1,942.05 |
| 7 | 516.56 |
| 8 | 979.28 |
| 9 | 241.83 |
| 10 | 1,494.21 |
| 11 | 1,515.05 |
| 12 | 1,017.23 |

Objective: To present the overall business performance for the year 2023

Methodology: Sum up the total price of "COMPLETED" transactions from the Billing table, grouping the results by month

Findings:

- Monthly revenue fluctuated, peaking in May-June
- The remaining months has relatively lower revenues, with April, July and September as the lowest three

4.View2: Monthly Top 5 Sold Dishes of 2023

```
CREATE VIEW MonthlyTop5Dishes AS
WITH Temp AS (
    SELECT MONTH(OD.CreateTime) AS [Month], DishName + ': ' + CAST(SUM([Count]) AS VARCHAR) AS DishData,
           RANK() OVER(PARTITION BY MONTH(OD.CreateTime) ORDER BY SUM([Count]) DESC) AS [Rank]
    FROM dbo.OrderDetail OD
    JOIN dbo.ShoppingCartItem SHC
    ON OD.ShoppingCartItemId = SHC.ShoppingCartItemId
    JOIN dbo.Dish D
    ON SHC.DishId = D.DishId
    WHERE OD.Status = 'COMPLETED'
    GROUP BY MONTH(OD.CreateTime), SHC.DishId, DishName
)
SELECT [Month], STRING_AGG(DishData, ', ') AS [Top5SoldDishes(DishName: SoldQuantity)]
FROM Temp
WHERE Rank <= 5
GROUP BY [Month];
```



Objective: To present the most popular dishes for each month

Methodology:

- Sum up the count of "COMPLETED" dishes from the orderDetail
- Group by month and dishId
- Organize data in horizontal format

Findings:

- Dishes are generally popular all the year round, peeking at Dec
- No dishes ordered in April and Sept.
- Kung Pao Shrimp is the most popular of all

| 123 Month | ABC Top5SoldDishes(DishName: SoldQuantity) |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Spring Onion Pancake: 8, Braised Pork: 5, Hot and Spicy Beef: 5, Sichuan Noodles: 3, Coconut Tapioca Pudding: 3 |
| 2 | Sweet and Sour Fish: 4, Cantonese Chow Mein: 3, Shrimp with Lobster Sauce: 3, Peking Duck: 3, Braised Pork: 3, Sichuan Noodles: 3 |
| 3 | Eggplant with Garlic Sauce: 6, Mango Sticky Rice: 6, Tom Yum Soup: 6, Spicy Hot Pot: 5, Szechuan Shrimp: 4 |
| 5 | Kung Pao Shrimp: 6, Hunan Pork: 5, Hot and Spicy Beef: 5, Sweet and Sour Pork: 5, Hot and Sour Soup: 4, Crispy Shrimp Wontons: 4, Chinese Corn Soup: 4 |
| 6 | Kung Pao Shrimp: 10, Peking Duck: 5, Dim Sum: 5, Spring Onion Pancake: 5, Tom Yum Soup: 4 |
| 7 | Wonton Soup: 3, Sichuan Noodles: 2, Tom Yum Soup: 2, Hunan Pork: 2, Szechuan Shrimp: 2, Coca Cola: 2 |
| 8 | Hot and Spicy Beef: 6, Kung Pao Chicken: 5, Crispy Shrimp Wontons: 5, Sweet and Sour Fish: 4, Shrimp with Lobster Sauce: 3, Mango Sticky Rice: 3, Mapo Tofu: 3 |
| 10 | Kung Pao Shrimp: 7, Mongolian Beef: 6, Lychee Iced Tea: 5, Shrimp with Lobster Sauce: 5, Sweet and Sour Fish: 4 |
| 11 | Peking Duck: 5, Szechuan Shrimp: 4, Wonton Soup: 4, Kung Pao Chicken: 4, Spring Onion Pancake: 4 |
| 12 | Mango Sticky Rice: 6, Spring Onion Pancake: 6, Sweet and Sour Pork: 3, Coconut Tapioca Pudding: 3, Chinese Corn Soup: 3, Spicy Hot Pot: 3, Lychee Iced Tea: 3, Sweet and Sour Fish: 3 |

4.View3: Monthly Top 5 Sold Setmeals of 2023

```
CREATE VIEW MonthlyTop5Setmeals AS
WITH Temp AS (
    SELECT MONTH(OD.CreateTime) AS [Month], SetmealName + ' : ' + CAST(SUM([Count]) AS VARCHAR) AS SetmealData,
           RANK() OVER(PARTITION BY MONTH(OD.CreateTime) ORDER BY SUM([Count]) DESC) AS [Rank]
    FROM dbo.OrderDetail OD
    JOIN dbo.ShoppingCartItem SHC
    ON OD.ShoppingCartItemid = SHC.ShoppingCartItemid
    JOIN dbo.SetMeal S
    ON SHC.SetmealId = S.SetmealId
    WHERE OD.Status = 'COMPLETED'
    GROUP BY MONTH(OD.CreateTime), SHC.SetmealId, SetmealName
)
SELECT [Month], STRING_AGG(SetmealData, ', ') AS [Top5SoldSetmeals(SetmealName: SoldQuantity)]
FROM Temp
WHERE Rank <= 5
GROUP BY [Month];
```



most popular setmeals for

| 123 Month | ABC Top5SoldSetmeals(SetmealName: SoldQuantity) |
|-----------|------------------------------------------------------------------------------------------------|
| 1 | Fresh A: 6, Beginnings A: 6, Feast: 4, Special A: 3, Farewell A: 3, Farewell B: 3 |
| 2 | Farewell C: 6, Farewell A: 4, Special A: 4, Kickoff: 3, Farewell B: 2, Fresh B: 2, Fresh A: 2 |
| 3 | Beginnings A: 3, Feast: 2, Special B: 1, Kickoff: 1, Fresh A: 1, Fresh B: 1 |
| 4 | Farewell B: 3, Kickoff: 3, Feast: 3, Special B: 2, Farewell C: 2, Farewell A: 2 |
| 5 | Fresh A: 3, Farewell A: 3, Farewell B: 2, Feast: 2, Beginnings A: 1, Farewell C: 1, Fresh B: 1 |
| 6 | Special B: 3, Kickoff: 2, Beginnings A: 1, Fresh B: 1 |
| 7 | Farewell A: 2, Special A: 1, Farewell B: 1 |
| 8 | Fresh A: 3, Farewell C: 1 |
| 9 | Kickoff: 3, Fresh A: 2, Fresh B: 1, Feast: 1 |
| 10 | Special B: 3, Kickoff: 1, Special A: 1 |
| 11 | Farewell A: 2, Special A: 1, Fresh B: 1, Fresh A: 1, Feast: 1, Beginnings A: 1 |
| 12 | Farewell C: 3, Beginnings A: 2, Kickoff: 2, Farewell B: 1, Special A: 1 |

Methodology:

- Sum up the count of "COMPLETED" setmeals from the orderDetail
- Group by month and setmealid
- Organize data in horizontal format

Findings:

- Semeals are most popular in Jan-May
- Semeals are no so popular in Jun-Dec
- Fresh and Farewell are the most popular two types of setmeal

Agenda

- 1.Fangying Tien: Design Document
- 2.Yiyun Lyu: ERD
- 3.Jingzhi Zhang: DDL
- 4.Yaqing Peng: Views
- 5.Yi-Chieh Huang: Visualization

Team 9 Restaurant Sales Dashboard

Total Sales
\$15,434.46

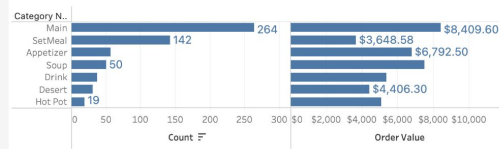
Order Numbers
105

Customer Numbers
10

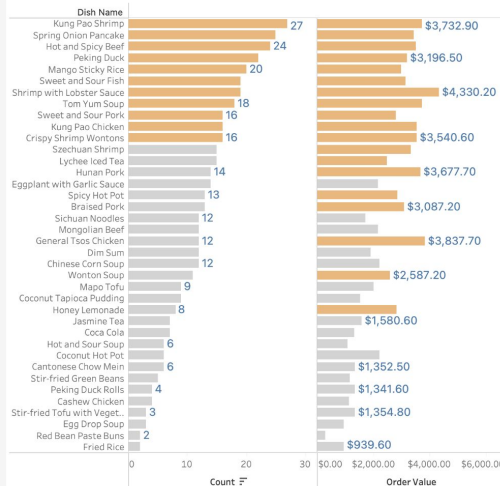
Monthly Order Numbers & Value



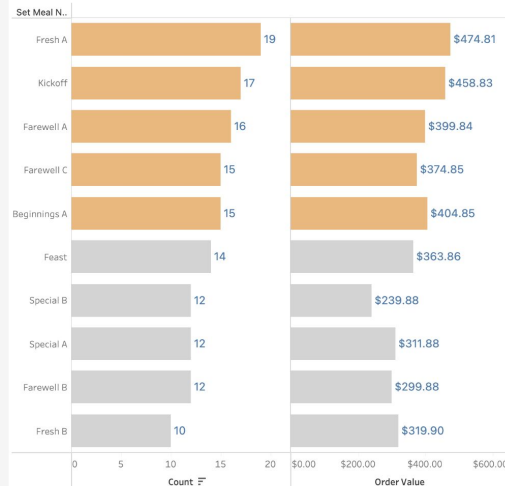
Best Selling Category



Best Selling Dish



Best Selling SetMeal



5. Interactive Visualizations

- Total View in Sales Values , Orders, and Customers
- Monthly Orders Sales Values
- Best Selling Category
- Best Salling Dish
- Best Selling SetMeal

Q&A Time



Thanks for listening!

Do you have any questions for us?