

Group 3:

Studio 3 Floral Events & Styling

Database Design



BY
SIQI ZHANG
MINH NGUYEN
MICHAEL ARVAN
YUNING LIU

1. INTRODUCTION:

In the modern world nowadays, a structured system with database is an essential for any form of businesses that wish to thrive, from retails to service, large corporations to family owns, or small local business to national-wide ones. For a local flower boutique, this standard stays the same: it needs more than just blooming flowers to flourish the business. Thereby, it is crucial for any flower shop to not only having beautiful flowers, but also a well-designed and properly operated database system to optimize the business flows and enhance customers satisfaction. Our mission is to become the ribbon for your bouquet, to create a simple yet elegant database system for small flower shops that able to fully optimize the sales, flowers delivery, supply control and customer experience by using information based on flowers, shipment details and customers.

2. PROBLEM STATEMENT:

Our client Studio 3 Floral Events & Styling is a local family owned business that have been in business for over 25 years in the Ballard region of Seattle WA. Studio 3 Floral provides several services like a location to be rented for events, styling for the events and flower delivery. Studio 3 Floral is owned by Pam and John, they wish to hand the family business over to there only son Mark. Mark has worked with his parents for 10 years and has a good handle of the business. However, Mark wants to reorganize how the family business will be ran since his parents want to retire. Mark want and need a platform where all his data can be access as well as an online present. We at Data Me Up, are working with Mark to learn his family needs and help him establish a database that will be as elegant as the work he and his family do.

There are four main parts in our database: ***customer records, order handling, supply management and warehousing management***. For ***customer records***, we design the system that each customer must register for membership and login before placing order. In this way, the floral store can not only keep track of customer's information, but also track their shopping behaviors through shopping cart. When customer adds items in their shopping cart, the database will record the product information, quantity, adding time, and modifying time of the items. After combining with order information, the floral store can analysis how long the customer will place the order after adding items in the cart, and for what price, how frequency they place the order, what time of the

day people are most inclined to review website and buy flowers, etc. Then, the florist can modify their marketing strategy based on these shopping behavior, for example, if one customer always buy the flower after discounting, the florist can provide random popup promotion code after the customer add items to motivate them placing the order. The database also can record membership status of each customer, which is another way to help florist improve their business strategy. Since Mark's business includes ***event handling and flower delivery***, so we design the database for these two things separately. For event handling, the florist can keep track of each event and manager who responsible for that event. And for flower delivery, the detailed shipping information and shipment process can be recorded for each order. If there is anything wrong with the delivery, it would be easier for the florist to track and solve the problem for customers. After buying the product, customer can write the review and rate the product which will also be recorded in our database. If the rating of one product is always not good, the ***supplier information*** can be found through ***product information***, and the floral store can change the suppliers who supplied that product. The ***warehouse*** is not only can record warehouse information, but also can supervise the suppliers. After supplier delivers the products to the warehouse, the employee of the warehouse can give the review and ratings of the suppliers and these information will also be recorded in the database. The rating will be based on the quality and delivery accuracy (including delivery time and quantity) of each supplier. If the ratings of one supplier is always not good, the florist can detect the problem and change the supplier. Both the customer review and employee review can help the florist modified their business and control the quality of products.

In conclusion, our database keep track of every part of the floral business needs, which can help Studio 3 Floral explore the potential business opportunities, satisfy customer needs, make appropriate decisions and increase their competitiveness.

3. SQL QUERIES

3.1 Find the customer buying the most expensive combo in a single time.

This query will find all of the customers who purchased the most expensive flower options in order to estimate the market demand and elasticity for that certain option. Since this query returns the customers who brought the most expensive option, the price will only return the highest price.

SQL Codes:

SELECT

c.cust_id AS 'CustomerID',

CONCAT(c.f_name, ' ', c.l_name) AS 'Customer Name',

o.total_price AS 'Product of Highest Price'

FROM

cust_info c,

order_info o

WHERE c.cust_id = o.cust_id

AND o.total_price = (

SELECT

MAX(o.total_price)

FROM

order_info o);

Subset of Data:

Customer ID	Customer Name	Product of Highest Price
16	Paula Diaz	3499.99
24	Ruby Rogers	3499.99

29	Maria Walker	3499.99
----	--------------	---------

3.2 Show all people who received the flowers instead of the customers.

This query will define customers who are purchasing flowers as a gift, either business or personal purposes, and retrieving the estimated arrival time for orders that have estimated arrival time, which allows the business owner to identify the seasonal shopping trends of their customers, or in this specific flower boutique, most flower gifted around Feb 14th, which is Valentine Day.

SQL Codes:

SELECT

concat(c.f_name,' ',c.l_name) AS 'Customer Name',
concat(s.recipient_f_name,' ',s.recipient_l_name) AS 'Recipient Name',
s.recipient_f_name AS 'Recipient First Name',
s.recipient_l_name AS 'Recipient Last Name',
s.est_arrive_time AS 'Estimated Arrival Time'

FROM

cust_info c,
order_info o,
shipping s

WHERE s.ship_id = o.ship_id

AND o.cust_id = c.cust_id
AND c.f_name != s.recipient_f_name
AND c.l_name != s.recipient_l_name
AND s.est_arrive_time > 0;

Subset of Data:

Customer Name	Recipient Name	Estimated Arrival Time
Benjamin Russell	Cheryl Scott	2020-02-14
Benjamin Russell	Paula Diaz	2020-02-12
Benjamin Russell	Son Tung	2020-02-16
Benjamin Russell	Theresa Lee	2020-02-06
Brenda Robinson	Britney Spear	2019-07-21
Carol Murphy	Ruby Rogers	2019-04-18
Carol Murphy	Carolyn Hayes	2020-02-19
Carol Murphy	Anne Russell	2020-02-09
Carol Murphy	Clark Ken	2020-03-10
Diana Peterson	Maria Walker	2020-02-09
Diane Evans	Dwayne Johnson	2020-02-27
Frances Young	Roger Roberts	2020-01-29
Joe Robinson	Diana Peterson	2020-02-28
Joe Robinson	Ralph Flores	2020-01-24
Joe Robinson	Jack Alexander	2020-02-23

Joe Robinson	Melissa King	2019-11-26
Lois Walker	Brenda Robinson	2020-02-23
Lois Walker	Joe Robinson	2020-01-17
Lois Walker	Diane Evans	2019-12-18
Lois Walker	Benjamin Russell	2018-10-09
Lois Walker	Minh Nguyen	2020-02-19
Lois Walker	Nancy Baker	2020-02-09
Lois Walker	Carol Murphy	2020-02-20
Nancy Baker	Carol Edwards	2019-11-27
Nancy Baker	Matthew Turner	2020-02-14
Nancy Baker	Adam Lavine	2020-02-14
Patrick Bailey	Julia Scott	2020-01-29
Patrick Bailey	Thomas Lewis	2020-02-15
Ralph Flores	Brenda Butler	2020-02-25

3.3 Show all next-day delivery orders.

This request allows the store owner to understand the urgent needs of customers (how many people request a next-day delivery), hence decide a better shipping option as well as an appropriate inventory system in order to optimize the customer purchasing experience and minimize the risk of stock out or failure to prepare the order on time.

SQL Codes:

SELECT

CONCAT(c.f_name, ' ', c.l_name) AS 'Customer Name',

o.order_id AS 'Order ID',

c.last_order_time AS 'Last order time',

s.est_arrive_time AS 'Estimated Arrival Time'

FROM

cust_info c,

order_info o,

shipping s

WHERE c.cust_id = o.cust_id

And o.ship_id = s.ship_id

And YEAR(c.last_order_time) = YEAR(s.est_arrive_time)

AND MONTH(c.last_order_time) = MONTH(s.est_arrive_time)

AND DAY(c.last_order_time) +1 = DAY(s.est_arrive_time);

Subset of Data:

Customer Name	Order ID	Last Order Time	Estimated Arrival Time
Lois Walker	101	2018-05-10	2018-05-11

3.4 Overview of total sales amount for each year

Create a view that shows the total sales amount for each year and compared with the next year. This view shows the sales years since the flower store opened, current sales amount which refers to the sales amount for that year, and next year sales amount which can be compared with the current sales amount.

SQL Codes:

CREATE VIEW SalesYearComparison AS

SELECT

YEAR(o.order_date) AS 'Sales Year',

SUM(d.price_after_disc) AS 'Current Sales Amount',

SUM(d.sale_price) AS 'Sales Amount Before Discount',

LEAD(sum(d.price_after_disc), 1, NULL)

OVER (order by o.order_date) AS 'Next Year Sales Amount',

LEAD(sum(d.sale_price), 1, NULL)

OVER (order by o.order_date) AS 'Next Year Sales Amount Before Discount'

FROM

order_info o,

order_detail d

WHERE o.order_id = d.order_id

GROUP BY YEAR(o.order_date);

Subset of Data

Sales Year	Current Sales Amount	Next Year Sales Amount	Next Year Sales Amount Before Discount
2017	87.12	783.39	880.89
2018	783.39	9661.69	9870.74
2019	9661.69	15347.45	15555.69
2020	15347.45	NULL	NULL

3.5 Show the age group with the most popular product

This query shows the product name, product quantity which are sold since the store were opened, and age group (i.e. 60s, 70s, 80s), which help the florist to target their customer and match specific products for different age groups.

SOL Codes:

SELECT

p.product_name AS 'Product Name',

count(d.product_id) AS 'Product Quantity',

CONCAT(TRUNCATE(YEAR(CURDATE()) - YEAR(c.dob), -1),

'-',

TRUNCATE(YEAR(CURDATE()) - YEAR(c.dob),-1)+9) AS 'Age Group',

CASE

WHEN p.category = 'Event' THEN 'Event Service'

ELSE 'In-Home Delivery'

END AS 'Product Category'

FROM

cust_info c,

order_detail d,

product p

WHERE c.cust_id = d.cust_id

AND d.product_id = p.product_id

GROUP BY d.product_id, 'Age Group'

ORDER BY 'Product Quantity';

Subset of Data:

Product Name	Product Quantity	Age Group	Product Category
Rose	12	50-59	In-Home Delivery
Poeny	9	50-59	In-Home Delivery
Orchids	4	50-59	In-Home Delivery
Rose Bundle	8	50-59	In-Home Delivery
Orchids Bundle	5	50-59	In-Home Delivery
Tulips	7	50-59	In-Home Delivery
Buttercup Bundle	6	30-39	In-Home Delivery
Birthday Event Flowers (small)	1	20-29	Event Service
Birthday Event Flowers (medium)	1	50-59	Event Service
Birthday Event Flowers (large)	2	20-29	Event Service
Wedding Event Flowers (small)	1	30-39	Event Service
Wedding Event Flowers (medium)	1	40-49	Event Service
Wedding Event Flowers (large)	3	30-39	Event Service
Graduation Event Flowers	1	30-39	Event Service
Party Event Flowers (small)	4	50-59	Event Service
Party Event Flowers (medium)	1	40-49	Event Service
Party Event Flowers (large)	1	20-29	Event Service

3.6 Show the manager who has the highest review score.

This query shows the manager id, manager name, manager's specialty, and average review score for each manager, then select the manager with the highest review score. This query can help florist find the best performance manager and make encourage mechanism accordingly.

SQL Codes:

```
SELECT e.manager_id AS 'Manager ID',  
       CONCAT(e.f_name, ' ', e.l_name) AS 'Manager Name',  
       e.specialty AS 'Specialty',  
       ROUND(AVG(r.score),1) AS 'Average Review Score',  
       COUNT(r.score) AS 'Number of Reviews'  
  
FROM  
  
event_manager e,  
  
cust_review r,  
  
event_service s,  
  
order_info o,  
  
cust_info c  
  
WHERE e.manager_id = s.manager_id  
  
      AND s.service_id = o.service_id  
  
      AND o.cust_id = c.cust_id  
  
      AND c.cust_id = r.cust_id  
  
      AND o.order_type = 'event service'  
  
      AND o.order_status = 'Completed'  
  
GROUP BY e.manager_id  
  
ORDER BY AVG(r.score) DESC LIMIT 1;
```

Subset of Data

Manager ID	Manager Name	Specialty	Average Review Score	Number of Reviews
3	Morris Reynalds	Wedding	5	1

3.7 Find the lowest price of each product and the corresponding supplier?

This query will find the lowest price of each product and the supplier that we currently have in our system.

SQL Codes:

SELECT DISTINCT

w.product_id,

(SELECT

product_name

FROM

product p

WHERE p.product_id = w.product_id) AS 'Product Name',

MIN(w.warehousing_price) AS 'Min Price Of Product',

s.supplier_name AS 'Supplier Name'

FROM

warehousing AS w

JOIN supplier AS s ON w.supplier_id = s.supplier_id

GROUP BY product_id;

Subset of Data:

Product ID	Product Name	Min Price Of Product	Supplier Name
21	Rose	4.99	Rosy Rose
27	Buttercup Bundle	3.75	Rosy Rose
35	Party Event Flowers (small)	5.99	Rosy Rose

36	Party Event Flowers (medium)	3.75	ButterCup Blossoms
28	Birthday Event Flowers (small)	4.99	Tuesday's Farms
37	Party Event Flowers (large)	4	Flower Me Up
22	Poeny	3.87	Boss Blossoms
29	Birthday Event Flowers (medium)	5.75	Hi Hi Bye
23	Orchids	5.99	Thompson, John Randolph Jr
24	Rose Bundle	2.35	Five Star
30	Birthday Event Flowers (large)	3.87	Five Star
31	Wedding Event Flowers (small)	2.5	Jackson Millwork Co
32	Wedding Event Flowers (medium)	2.35	Ac Supply Co
25	Orchids Bundle	4.22	Tri Co
33	Wedding Event Flowers (large)	2.45	Tri Co
26	Tulips	1.99	Cowan & Kelly flower planting
34	Graduation Event Flowers	1.99	Mcrae, James L

3.8 How much of each product do we have in our warehouse?

This query will display the product id, product name and the amount of items we have in our warehouse.

SQL Codes:

SELECT DISTINCT

product_id AS 'Product ID',

(SELECT

product_name

FROM

product p

WHERE p.product_id = w.product_id) AS 'Product Name',

SUM(warehousing_amt) AS Amount

FROM

warehousing as w

GROUP BY product_id

ORDER BY SUM(warehousing_amt) DESC;

Subset of Data:

Product ID	Product Name	Amount
21	Rose	490
22	Poeny	350
24	Rose Bundle	258
27	Buttercup Bundle	140
23	Orchids	60
26	Tulips	56

25	Orchids Bundle	28
28	Birthday Event Flowers (small)	17
32	Wedding Event Flowers (medium)	17
33	Wedding Event Flowers (large)	15
34	Graduation Event Flowers	14
30	Birthday Event Flowers (large)	13
36	Party Event Flowers (medium)	12
31	Wedding Event Flowers (small)	10
29	Birthday Event Flowers (medium)	9
37	Party Event Flowers (large)	8
35	Party Event Flowers (small)	6

3.9 How many suppliers do we have per product?

This query will display the number of suppliers we have for each product that are current in our system. This is important when we try to find more suppliers for a product or negotiate a product price with supplier for a product that we have lot of suppliers for.

SOL Codes:

SELECT

product_id AS 'Product ID',

(SELECT

product_name

FROM

product p

WHERE p.product_id = w.product_id) AS 'Product Name',

COUNT() AS 'Number of Suppliers per product'*

FROM

warehousing as w

GROUP BY product_id;

Subset of Data:

Product ID	Product Name	Number of Suppliers per product
21	Rose	4
22	Poeny	2
23	Orchids	1
24	Rose Bundle	7
25	Orchids Bundle	1
26	Tulips	2
27	Buttercup Bundle	5
28	Birthday Event Flowers (small)	3
29	Birthday Event Flowers (medium)	2
30	Birthday Event Flowers (large)	2

31	Wedding Event Flowers (small)	2
32	Wedding Event Flowers (medium)	3
33	Wedding Event Flowers (large)	3
34	Graduation Event Flowers	2
35	Party Event Flowers (small)	2
36	Party Event Flowers (medium)	2
37	Party Event Flowers (large)	2

3.10 How many members joined our website per year?

This query will display the number of members that have joined the flower store website.

SQL codes:

SELECT

YEAR(mem_since) AS YEAR,

COUNT() AS 'Num of Member Joined Per Year'*

FROM

cust_info

GROUP BY YEAR(mem_since)

ORDER BY COUNT() DESC;*

Subset of Data:

YEAR	Num of Member Joined Per Year
2016	11
2017	7
2015	4
2018	3
2014	3
2020	1
2019	1

3.11 STORED PROCEDURE

3.11.1 Get sales amount of specific product in a given year and month.

This stored procedure allows the florist check their total sales amount for a given year and month, and the product name accordingly. The florist can check the sales status for specific product at specific period.

DELIMITER \$\$

CREATE PROCEDURE ProductSalesByYearMonth

(IN ProductName Varchar(60),

IN SalesYear INT,

IN SalesMonth INT)

BEGIN

SELECT

p.product_id AS 'Product ID',

p.product_name AS 'Product Name',

p.category AS 'Category',

YEAR(o.order_date) AS 'Sales Year',

MONTH(o.order_date) AS 'Sales Month',

o.order_type AS 'Order Type',

sum((o.total_price + s.shippment_fee)) AS 'Total Sales Amount'

FROM

product p,

order_info o,

shipment_fee s,

order_detail d

WHERE

```

p.product_id = d.product_id
AND o.order_id = d.order_id
AND o.shipment_fee_id = s.shipment_fee_id
AND p.product_name = ProductName
AND YEAR(o.order_date) = SalesYear
AND MONTH(o.order_date) = SalesMonth

GROUP BY d.product_id;

END$$

DELIMITER ;

```

Test trial for getting sales amount for Rose in Feb 2020:

```
CALL ProductSalesByYearMonth('Rose', 2020, 02);
```

Output:

Product ID	Product Name	Category	Sales Year	Sales Month	Order Type	Total Sales Amount
21	Rose	Flower	2020	2	delivery	104.97

3.11.2 Total order amount placed by customer

This store procedure will allow any clerk to pull up information about any customer based on their customer id, and pull out the total order number that customer placed.

```

DELIMITER $$
CREATE PROCEDURE FindCustomerInfo
    (IN ID int)
BEGIN
SELECT
    c.cust_id AS 'Customer ID',
    CONCAT(c.f_name, ' ', c.l_name) AS 'Customer Name',
    COUNT(*) AS 'Total Order Number'

```

```

FROM
    cust_info c,
    order_info o
WHERE
    c.cust_id = o.cust_id
    And o.cust_id = ID
GROUP BY c.cust_id;
END$$
DELIMITER ;

```

Test trial for getting total shipping fee for cust_id = 1:

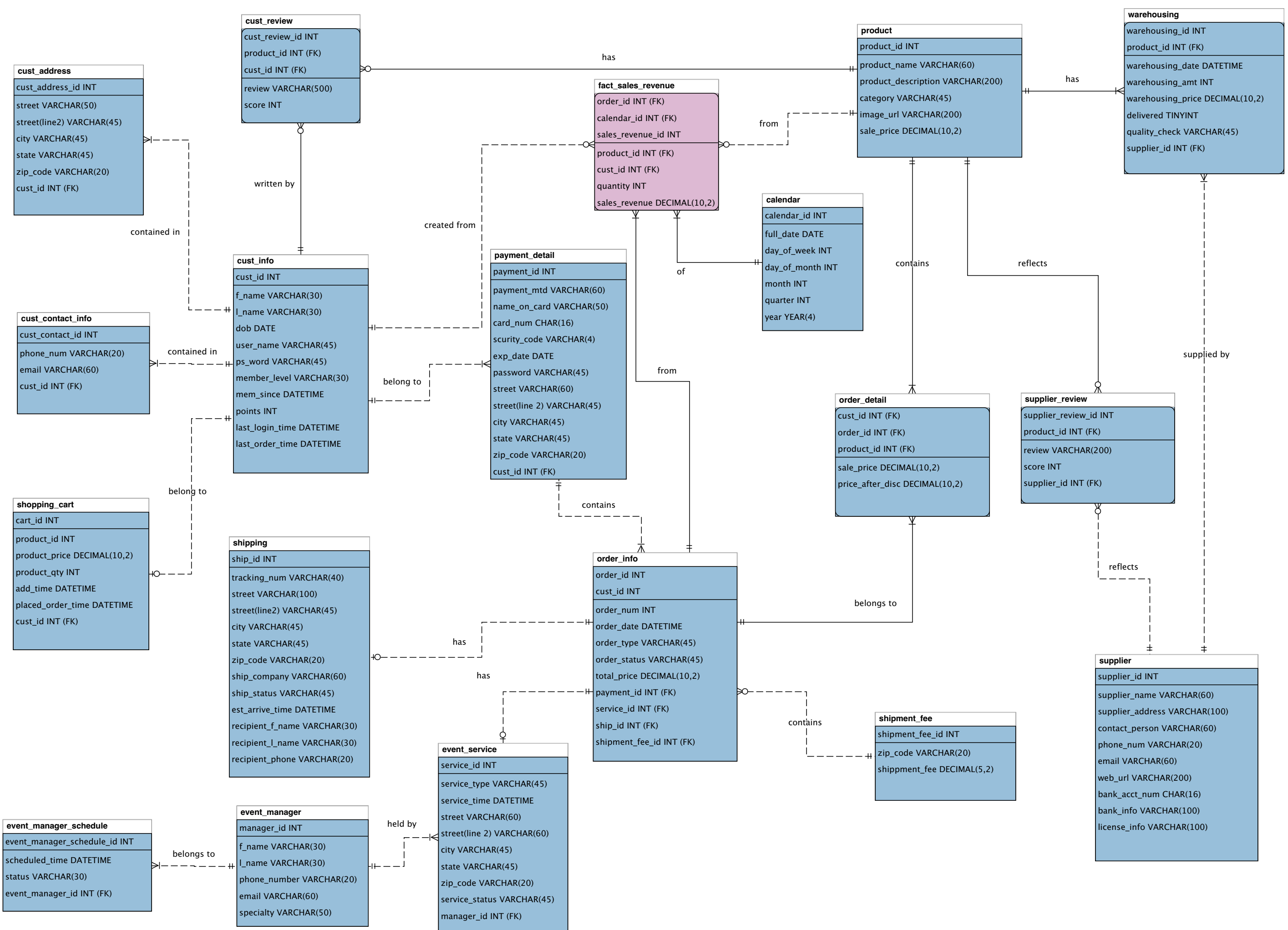
CALL FindCustomerInfo(1);

Output:

Customer ID	Customer Name	Total Order Number
1	Lois Walker	8

4. ER DIAGRAM AND BUSINESS RULES

4.1 ER Diagram (Fact Table Included)



4.2 Business Rules

4.2.1 Customers:

Customer DoB must be in the yyyy-mm-dd format. This will allow the flower boutique to fully retrieve the customer's age, birthdate, and related special events for promotions and flower options. Also, as this flower boutique is a growing local business, all of the customer's address and phone number must be within the United States. This will restrict any invalid address or any phone number with incorrect format or outside of the U.S. Customers are welcome to make any flower reservation or orders in bundle ahead of time, but no longer than 1 year. Because flowers are seasonal dependable, thereby, if anything out of the sudden happen to next year season that cannot be foreseen, the flowers might be damaged or not ready to harvest. This will resulted in a failure order for the customers and negatively impact the store's reputation. Also, customer's feedbacks and reviews are more than welcome, yet any comment that is longer than 500 words is prohibited. For such concern, customers might want to contact the store owners or managers directly for the best service.

4.2.2 Orders:

To remain a competitive advantage on the market, shipping fee per customer per order must be kept in a reasonable price, no matter the size of the order. For some special event where an order might cost more than \$1,000, the store owner can increase the shipping fee but only up to 300 USD. For orders that are in the cart, the order placed time must be after the time the items being placed inside the cart, or in other words, $\text{place_order_time} > \text{add_time}$. This means customers can feel free to place any items in the cart, yet they cannot put in and order at once. This will allow the store owners to observe customers' shopping habits through the shopping cart hence come up with

a proper marketing strategy. Lastly, the estimate arrival time for any orders must be larger than the last order time.

4.2.3 Products:

For any products of the flower boutique, a description will be required and will be demonstrated to the customers before purchasing the flowers. However, this description should not exceed 200 words in length, as a long and overwhelmed description would make the reader feel bored. Moreover, a short yet informative description should be the key strategy here. A product might carries a certain amount of discount in percentage, yet the discounted price should not be more than a half of the product's original price. This mechanism ensures that the store owner will not be suffering from any loss per item in the future. Also, product review is based on a 5 stars scale, so the product's score maximum will be 5.

4.2.4 Warehousing:

To maintain an economic benefit to the flower store, warehouse price should not be larger than 10USD, any price that is higher than that should not be accepted and need to discuss or review with the manager or store owners. Warehousing date should be smaller than estimate arrival time for any orders. This means any products or orders will first be transferred from the suppliers to the warehouse, not directly from the suppliers. This procedure will ensure that any products of the flower store will go through the quality control, with maximum value of 10, inside the warehouse for the optimal flower's quality for the customers.

4.2.5 Suppliers:

For now, the flower shop will only be focused in the domestic flowers type, meaning the suppliers would be from the U.S. Hereby, similar to customers, suppliers address and phone number must be in the U.S. Supplier review rating system will be based on the scale of 10, which mean the maximum value of supplier review score is 10.

5. ACCESS TO MYSQL DATABASE IN THE CLOUD

Host: mmcp5910team03.cillrbubr4qu.us-east-1.rds.amazonaws.com

Username: admin

Password: team3333

6. ANALYTICAL SQL QUERIES AND TABLEAU GRAPHS

6.1 Business Analysis and SQL Queries

Product Category

- **Popularity Analysis:** There are three product categories “single flower”, “packaged flower” and “event services”. After combining category , sales amount and total revenue together, we can find which product can provide the highest revenue by year, quarter, month or by day. The florist store can provide some promotions and increase sales on these popular products to further increase sales revenue.
- **Quality Analysis:** Product quality is mainly based on average customer reviews scores. The higher the score, the more satisfied the customers. Also, for some lower score products or category, the florist should take measures on supplier management and warehousing storage management to make sure the quality of the product.
- **Diversity Analysis:** From the category analysis, the florist store can find whether they need to increase the diversities of the product category to increase their revenue in the future.

Market

Popularity Analysis: The florist store can directly see the revenue mainly comes from which area. Whether they have market potential in some specific area. The sale trend in each city, and whether they need to use some business strategies, like launching advertising, provide promotions, etc., to increase the sales revenue.

Market Potential Analysis: For now, their business mainly focus on Seattle, Bellevue, Redmond, Kirkland, and Sammamish area. They can explore more potential markets around these areas to increase their revenue. For some areas creating low revenue, the florist store can take some targeted measures or do customer surveys for their further business.

Combine category and market together

- **Product analysis on whole market:** The florist store can find out which product category can bring the highest revenue in which area. The customer in which city more prefer to buy which product in which category. And they also can see the revenue trend of each product category across different cities, which can help them further understanding their current market based on their product.
- **Forecast the market trend and target the problem:** When they have a specific understanding about their current market, they can analysis the customer purchasing behavior in different cities across different product and forecast the further needs, which can help them manage the warehousing.

Based on these analysis needs, we created several SQL queries and visualized them in Tableau to provide a more straightforward way to show our analysis.

6.2 SQL Queries for Analytical Analysis

Query 1

```
SELECT
    COUNT(o.order_id) AS 'Order Amount',
    SUM(o.total_price) AS 'Revenue',
    YEAR(o.order_date) AS 'Sales Year',
    QUARTER(o.order_date) AS 'Sales Quarter',
    s.city AS 'City'
```

```
FROM
    order_info o,
    order_detail d,
    shipping s
WHERE o.order_id = d.order_id
    AND o.ship_id = s.ship_id
GROUP BY YEAR(o.order_date), QUARTER(o.order_date), s.city
ORDER BY YEAR(o.order_date), QUARTER(o.order_date), s.city;
```

Query 2

```
SELECT
    SUM(o.total_price) AS 'Revenue',
    ROUND(AVG(r.score),1) AS 'Average Review Score',
    COUNT(o.order_id) AS 'Order Amount',
    p.category AS 'Product Category'
FROM
    order_info o,
    order_detail d,
    cust_review r,
    product p
WHERE o.order_id = d.order_id
    AND d.product_id = p.product_id
    AND p.product_id = r.product_id
GROUP BY p.category;
```

Query 3

```
SELECT
    SUM(o.total_price) AS 'Revenue',
    YEAR(o.order_date) AS 'Sales Year',
    QUARTER(o.order_date) AS 'Sales Quarter'
```

```
FROM
    order_info o
GROUP BY YEAR(o.order_date), QUARTER(o.order_date)
ORDER BY YEAR(o.order_date), QUARTER(o.order_date);
```

Query 4

```
SELECT
    YEAR(o.order_date) AS 'Sales Year',
    SUM(d.price_after_disc) AS 'Current Sales Amount',
    SUM(d.sale_price) AS 'Sales Amount Before Discount',
    LAG(sum(d.price_after_disc), 1, NULL)
    OVER (order by o.order_date) AS 'Previous Year Sales Amount',
    LAG(sum(d.sale_price), 1, NULL)
    OVER (order by o.order_date) AS 'Previous Year Sales Amount Before Discount'
FROM
    order_info o,
    order_detail d
WHERE o.order_id = d.order_id
GROUP BY YEAR(o.order_date)
ORDER BY YEAR(o.order_date);
```

We created two interactive dashboard, and each dashboard contains three graphs. You can click each part in each graph to have a further look.

Link 1:

<https://public.tableau.com/profile/siqi.zhang1612#!/vizhome/FloristStoreSalesRevenueAnalysis/Dashboard1?publish=yes>

Link 2:

<https://public.tableau.com/profile/siqi.zhang1612#!/vizhome/FloristStoreSalesRevenueAnalysisbyCity/Dashboard2?publish=yes>

7. GitHub Link

Here is GitHub link of our project:

<https://github.com/siqizhang1130/Florist-Store-Database-Design/tree/master>