JML Garden Produce

Database Design Project

Group 2: Victor Li, Tinsley Zhu, Kaitlyn Li, Yikang Li, Candice Hu, Alex Tomaso, Joyce Lo, Rupert Kay, Sasha Itkis, Brandon Luo
Agenda

- Client Summary
- Database Implementation
- Queries and Models
- Normalization
Client Summary
JML Garden Produce is a distributor of Asian vegetables in South El Monte, CA connecting farms and producers to supermarkets and wholesale customers.

- 6 Employees
- ~30 shipments/day
- ~20,000 lbs of produce sold/day
Client Summary

Agenda | Client Summary | Database | Queries and Models | Normalization
Client Summary

Agenda | Client Summary | Database | Queries and Models | Normalization
Database Implementation
EER Diagram
Relational Schema
1. ProduceType(ProduceID, ProduceName, ProduceDescription, ShelfLife, Refrigerated, Category, Season)
2. Customer(CustomerID, CustomerName, ContactName, Email, Phone, LocationID)
3. Supplier(SupplierID, SupplierName, ContactName, Email, Phone, LocationID)
4. Order(OrderID, SalespersonSSN, Order_DateString)
   a. OrderFromCustomer(OrderID, CustomerID)
   b. OrderToSupplier(OrderID, SupplierID)
5. OrderContainsProduceType(ProduceID, OrderID, unitPrice, quantity)
6. Quote(QuoteID, SalespersonSSN, Quote_DateString)
   a. QuoteToCustomer(QuoteID, CustomerID)
   b. QuoteFromSupplier(QuoteID, SupplierID)
7. QuoteContainsProduceType(QuoteID, ProduceID, unitPrice)
8. Delivery(DeliveryID, DriverSSN, Delivery_DateString)
   a. DeliveryToCustomer(DeliveryID, OrderID)
   b. DeliveryFromSupplier(DeliveryID, SupplierID)
9. DeliveryTC_containsCustomerOrder(DeliveryID, OrderID)
10. DeliveryFS_containsSupplierOrder(DeliveryID\(^8\), OrderID\(^4\))

11. DeliveryLeg(DeliveryID\(^8\), LegNo., StartLocationID\(^12\), EndLocationID\(^12\))

12. Location(LocationID, StreetAddress, City, Country, Zipcode)

13. Warehouse(WarehouseID, LocationID\(^13\), Capacity, #_Of_fridge)

14. Equipment(EquipmentID, WarehouseID\(^15\))

15. Date(DateString)

16. Employee(SSN, Lname, MI, Fname, Bdate, Email, Phone, Address, WeeklyHours, Manager_SSN\(^{18b}\))
   a. Salesperson(SSN\(^{18}\), commission)
   b. Manager(SSN\(^{18}\), salary)
   c. Driver(SSN\(^{18}\), wage)
   d. ForkliftOperator(SSN\(^{18}\), wage)
   e. Other(SSN\(^{18}\), salary)

17. OperatorCanUseEquipment(SSN\(^{18d}\), EquipmentID\(^16\))

18. InventoryUnit(InventoryID, ProduceID\(^1\), Expiration_DateString\(^{17}\), DeliveryID\(^8\), SpaceID\(^{21}\))

19. Space(SpaceID, WarehouseID\(^{15}\), ShelfLevel, AisleNumber, Refrigerated_Or_Not)
MS Access Relationships
Queries and Models
Query 1: Route Optimization
Query 1: Motivation and Goal

What is the best way to make all our deliveries?

Motivation:

→ Routes can be created and scheduled ahead of time using the orders and locations from the customers to increase efficiency in terms of employee time and truck utilization

Methodology:

1. Find **deliveries** with SQL query
2. Minimize time costs per truck using linear programming
3. Return optimal delivery route
Query 1: SQL

Creates a table with locations of all customers with orders needing delivery

```
SELECT loc.StreetAddress, loc.City, loc.State, loc.ZipCode, SUM(ocpt.quantity) AS OrderSize
FROM OrderFromCustomer AS ofc, OrderContainsProduceType AS ocpt, Customer AS cus, Location AS loc
GROUP BY loc.LocationID, loc.StreetAddress, loc.City, loc.State, loc.ZipCode
HAVING loc.LocationID IN

(SELECT DISTINCT loc2.LocationID
FROM Location AS loc2, Customer AS cus2, OrderFromCustomer AS ofc2
WHERE ((loc2.LocationID)=[cus2].[LocationID]) AND ((cus2.CustomerID)=[ofc2].[CustomerID]) AND ((ofc2.OrderID) IN

(SELECT ofc3.OrderID
FROM ([OrderFromCustomer] AS ofc3)
WHERE ofc3.OrderID NOT IN

(SELECT DISTINCT dto4.OrderID
FROM [DeliveryTC_containsCustomerOrder] AS dto4)))));
```
### Query 1: SQL Results

#### Route Optimization

<table>
<thead>
<tr>
<th>StreetAddress</th>
<th>City</th>
<th>State</th>
<th>ZipCode</th>
<th>OrderSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>3828 N Peck Rd</td>
<td>El Monte</td>
<td>CA</td>
<td>91732</td>
<td>270</td>
</tr>
<tr>
<td>11850 E Valley</td>
<td>El Monte</td>
<td>CA</td>
<td>91732</td>
<td>180</td>
</tr>
<tr>
<td>8235 Garvey Av</td>
<td>Rosemead</td>
<td>CA</td>
<td>91170</td>
<td>69</td>
</tr>
<tr>
<td>1200 E Valley Bld</td>
<td>Alhambra</td>
<td>CA</td>
<td>91801</td>
<td>380</td>
</tr>
</tbody>
</table>
We use **Google API data** to find travel time between pairs of locations and uses **mixed-integer linear programming** in **AMPL** to calculate the set of routes that visits all locations with minimum total travel time.

```plaintext
param nvert;
param nroutes;
param maxcap;

set V = {1..nvert}; #number of nodes
set C = {1..nroutes}; #max number of routes

param w[i in V, j in V];
param d[i in V];

var x[i in V, j in V, c in C] binary;

minimize cost: sum{i in V, j in V, c in C} (x[i,j,c] * w[i,j]);

subject to

EdgesFormCycles (j in V, c in C): sum {i in V} (x[i,j,c]) - sum {k in V} (x[j,k,c]) = 0;

AllNodesVisited (j in V): sum {i in V, c in C} x[i,j,c] >= 1;

CycleLeavesWarehouse (c in C): sum {j in V} x[1,j,c] = 1;

CycleReturnsToWarehouse (c in C): sum {i in V} x[i,1,c] = 1;

CycleMaxCap (c in C): sum {i in V, j in V} x[i,j,c] * d[i] <= maxcap;
```
Query 1: Benefits

✓ Reduce travel time and expenditures
✓ Spend less time planning routes for drivers
✓ Positive environmental impact
Query 2: Customer Tracking
How can we track our customers most effectively?

**Motivation:**
- We are interested in seeing which of JML’s customers purchase the most goods over a time period, and would be responsive to certain promotions or discounts.

**Methodology:**
- Query customer past order information
  - Use metrics to determine customer behavior
  - Apply promotions and measure response
Query 2: SQL

Creates a table with customers’ previous order quantities, order prices, and order dates

```
SELECT c.CustomerName, c.CustomerID, p.ProduceName, op.Quantity, d.Date_String
FROM Customer AS c, OrderFromCustomer AS oc, Order AS o,
OrderContainsProduceType AS op, ProduceType AS p, Date AS d
WHERE c.CustomerID = oc.CustomerID AND d.Date_String = o.Order_DateString AND
p.ProduceID;
```

<table>
<thead>
<tr>
<th>CustomerName</th>
<th>CustomerID</th>
<th>ProduceID</th>
<th>Quantity</th>
<th>UnitPrice</th>
<th>Date_String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cool Food</td>
<td>3</td>
<td>1</td>
<td>10</td>
<td>$0.50</td>
<td>1/1/2017</td>
</tr>
<tr>
<td>Cool Food</td>
<td>3</td>
<td>2</td>
<td>20</td>
<td>$0.50</td>
<td>1/1/2017</td>
</tr>
<tr>
<td>Cool Food</td>
<td>3</td>
<td>5</td>
<td>39</td>
<td>$0.50</td>
<td>1/1/2017</td>
</tr>
<tr>
<td>Asian Food Store</td>
<td>1</td>
<td>14</td>
<td>50</td>
<td>$0.70</td>
<td>1/2/2017</td>
</tr>
<tr>
<td>Asian Food Store</td>
<td>1</td>
<td>8</td>
<td>60</td>
<td>$0.70</td>
<td>1/2/2017</td>
</tr>
</tbody>
</table>
Query 2: Benefits

✓ Understand customer behavior

✓ Understand key areas of business

✓ Take advantage of clustering
Query 3: Supplier Tracking
How can we assess the reliability and performance of suppliers?

**Motivation:**
- Save time spent in seeking quotations by optimizing our list of suppliers
- Predict the quotations of each supplier for different produce type and strengthen partnerships with the top three suppliers in each category to seek further discounts

**Methodology:**
- Find Suppliers’ quotes and prices
- Find trustworthy suppliers by performing linear regression
- Rank trustworthiness of suppliers from optimized list
Query 3: SQL

Creates a table with suppliers’ quotations and sale prices for different types of produce

```
SELECT os.SupplierID, ocpt.ProduceID, ocpt.unitPrice AS OrderUnitPrice, qcpt.unitPrice AS QuoteUnitPrice, o.Order_DateString
FROM (SELECT os.SupplierID, ocpt.ProduceID, ocpt.unitPrice, o.Order_DateString
    FROM OrderToSupplier s, OrderContainsProduceType ocpt, Order o
    WHERE s.OrderID = ocpt.OrderID, o.OrderID = s.OrderID)
    FULL OUTER JOIN (SELECT qs.SupplierID, qcpt.ProduceID, qcpt.unitPrice, q.Quote_DateString
    FROM QuoteFromSupplier qs, QuoteContainsProduceType qcpt, Quote q
    WHERE s.QuoteID = qcpt.QuoteID, q.QuoteID = s.QuoteID)
ON os.SupplierID = qs.SupplierID AND ocpt.ProduceID = qcpt.ProduceID AND o.Order_DateString = q.Quote_DateString
ORDER BY o.Order_DateString;
```
### Query 3: Supplier Tracking

<table>
<thead>
<tr>
<th>A.SupplierID</th>
<th>A.ProduceID</th>
<th>A.unitPrice</th>
<th>Order_Date</th>
<th>B.SupplierID</th>
<th>B.ProduceID</th>
<th>B.unitPrice</th>
<th>Quote_Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$0.50</td>
<td>1/1/2017</td>
<td>1</td>
<td>1</td>
<td>$2.57</td>
<td>1/1/2017</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>$0.50</td>
<td>1/1/2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>$0.50</td>
<td>1/1/2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>$0.00</td>
<td>1/2/2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>$0.70</td>
<td>1/2/2017</td>
<td>1</td>
<td>14</td>
<td>$2.20</td>
<td>1/2/2017</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>$0.60</td>
<td>1/2/2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>$0.70</td>
<td>1/2/2017</td>
<td>1</td>
<td>15</td>
<td>$3.35</td>
<td>1/2/2017</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$0.30</td>
<td>1/2/2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>$0.40</td>
<td>1/2/2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>$0.40</td>
<td>1/2/2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>$0.30</td>
<td>1/2/2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>$0.30</td>
<td>1/2/2017</td>
<td>4</td>
<td>11</td>
<td>$1.57</td>
<td>1/2/2017</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>$0.90</td>
<td>1/2/2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Query 3: Success Rate and Linear Regression

- % success rate of supplier i = # of quotations from supplier/ # of successful orders to suppliers
- Perform linear regression to predict quotations after extracting relevant weather data from NOAA database

\[
\text{Quote\_UnitPrice} = \beta_0 + \beta_1 \text{ supplier\_ID} + \beta_2 \text{ produce type} + \beta_3 \text{ weekday\_or\_not} + \beta_4 \text{ holiday\_or\_not} + \beta_5 \text{ weather\_type} + \beta_6 \text{ temperature} + \epsilon
\]

R code:

```r
lm(unit_price~as.factor(supplier_ID)+ as.factor(weekday_or_not) + as.factor(weather) + temperature + as.factor(holiday_or_not) + I(as.factor(produce_type) + QuoteUnitPrice), data= Suppliertracking)
```
Query 3: Benefits

✓ Choose best supplier for each produce type
✓ Prefer purchasing from suppliers that are more accurate
✓ Reduce price variability/uncertainty
Query 4: Inventory Management
How do we keep track of current inventory levels and restock appropriately?

Motivation:

➔ Excess inventory incur a carrying cost and worse, cannot be sold after the expiration date
➔ Stockouts (unfulfilled demands) are lost profit
➔ Hard for employees to intuitively understand which produce are selling well and which ones need more promotions

Methodology:

1. Use SQL queries to obtain produce’ expiration dates
2. Separately perform EOQ analysis
3. Make restock orders appropriately to maintain inventory
Query 4: SQL

Creates table with inventory units with corresponding type, location, and expiration

```
SELECT i.inventoryid, s.spaceid, i.expiration_datestring, p.produceid
FROM InventoryUnit AS i, Space AS s, ProduceType AS p
WHERE i.spaceid = s.spaceid AND i.produceid = p.produceid
ORDER BY i.expiration_datestring;
```
Query 4: EOQ Model

Optimal Quantity ($Q^*$):
\[ \sqrt{\frac{2 \times \text{orderingCost} \times \text{DailyDemand} \times 365\text{days}}{\text{AnnualInterestRate} \times \text{UnitCost}}} \]

Reorder Point ($R$):
\[ \text{Optimal Cycle Time (T*)} \times \text{Lead Time} \]

Optimal Cycle Time ($T^*$):
\[ \frac{\text{OptimalQuantity} (Q^*)}{\text{DailyDemand} \times 365\text{days}} \]
Query 4: Assumptions

➔ All variables (Daily Demand, ordering cost, unit cost, and interest rate) are known and constant

➔ Delivery Lead Time is 1 day

➔ 365 days a year
Query 4: Steps

- An order is placed at the \( \text{current date} + \frac{Q^*}{\text{daily demand}} \) or when a stockout occurs or inventory hits the expiration date, whichever occurs earlier.

- A stockout happens when the inventory of the day is less than the predicted daily demand, and the replenishing order is placed immediately, which will arrive with lead time of 1 day.

- If new inventory arrives before the old is depleted, the new inventory’s expiration date will be stored as Expiration Date #2, and the old produce will be sold before the new.
Query 4: Benefits

✓ Better restocking to fulfill demand and lower excess inventory

✓ Better manage produce expiration and reduce waste

✓ Save employee time looking for items
Query 5: Price Forecasting
How do predict supply, demand and pricing for our goods?

Benefits:

➔ Being able to predict changes in supply, demand and prices can benefit the company by knowing when to keep more inventory and can help make or save the company more money
➔ Could potentially negotiate contracts to lessen the impact of price fluctuations

Methodology:

1. Query historical price data
2. Utilize machine learning models
3. Predict supply and demand equilibrium
Query 5: SQL

Creates a table of produce types with previous order prices, quantities, customers, and dates

```sql
SELECT op.ProduceID AS ProduceID, op.unitPrice AS Unit_Price, op.quantity AS Quantity, oc.CustomerID, o.Order_DateString
FROM OrderContainsProduceType AS op, Order AS o, OrderFromCustomer AS oc
WHERE op.OrderID = o.OrderID AND oc.OrderID = o.OrderID
ORDER BY op.ProduceID, o.Order_DateString;
```
We use historical datasets of tomato PPI, weather data, macroeconomic factors to predict futures prices of tomatoes.

We use ARIMA to take into account trends such as seasonality along with other exogenous factors.
Query 5: Benefits

✓ Understand supply and demand of goods
✓ Predict prices to save money when buying in bulk
✓ Reduce losses from waste or missed sales
Normalization Analysis

➔ Our database is in 3rd Normal Form

➔ We did not need to make any modifications
Questions?