The Online Juice Shop Database Project

Yan Wang
Jingnan Qi
Jing Kunzler
Xiang Pan

The following materials document the design and development of a database application to support a small online juice shop. The project begins with a description of the business and proceeds through conceptual (E-R) modeling, logical (Relational) modeling, Physical modeling and finally implementation of a database application. Notes are provided at the end of each section to explain some specific features of the steps being carried out.

I. Business Scenario

We are an online healthy juice bar. We have been using Excel spreadsheets to maintain logs of suppliers, products, orders, and customers' information, and we would like to create a new database management system that dynamically communicate between sheets.

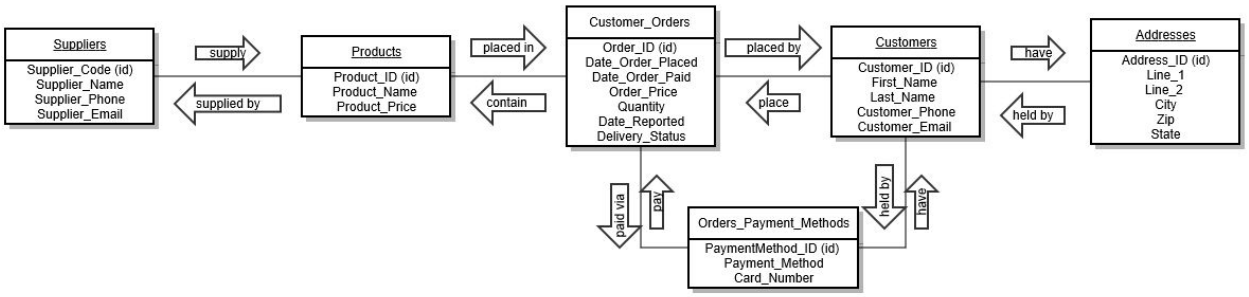
Our business model is as such: we receive bottles of juice from our suppliers nationwide, and we ship to individuals in the US. We need to track the supplier information such as names and phone numbers, product information such as varieties and price, order information such as order date, quantity, and sales amount, and customer information such as address, contact information, and payment methods. And we would like to inquire answers to other dynamic business questions such as which are our loyal customers with a steady relationship, and which have decreased orders from us, to identify risks and control issues.

Commentary:

Based on the above description, construct an Entity Relationship model using UML notation that will capture all of the business' data needs.

II. ER Model using UML Notation

Based on the above description, we develop the following Entity Relationship model using UML notation.



Relationship Sentences

One **Suppliers** *must be* supplying one or more **Products**
One **Products** *must be* supplied by one or more **Suppliers**

One **Products** *may be* placed in one or more **Customer_Orders**
One **Customer_Orders** *must* contain one or more **Products**

One **Customers** *may be* placing one or more **Customer_Orders**
One **Customer_Orders** *must be* placed by one and only one **Customers**

One **Customers** *must be* having one or more **Addresses**
One **Addresses** *must be* held by one or more **Customers**

One **Customer_Orders** *must be* paid via one **Orders_Payment_Method**
One **Orders_Payment_Method** *may be* paying one or more **Customer_Orders**

One **Customers** *may be* having one or more **Orders_Payment_Method**
One **Orders_Payment_Method** *must be* held by one and only one **Customers**

III. Conversion to Relational Model

Suppliers(Supplier_Code (key), Supplier_Name, Supplier_Phone, Supplier_Email)
Products(Products_ID (key), Product_Name, Product_Price)
Customers(Customers_ID (key), First_Name, Last_Name, Customers_Phone, Customers_Email)
Addresses(Address_ID (key), Line_1, Line_2, City, Zip, State)

Customer_Orders(Order_ID (key), Customer_ID (fk), PaymentMethod_ID (fk), Date_Order_Placed, Date_Order_Paid, Order_Price, Date_Reported, Delivery_Status, Product_ID, Quantity)
Orders_Payment_Method(PaymentMethod_ID (key), Order_ID(fk), Payment_Method_Code, Card_Number)

IV. Normalization

1) **Suppliers**(Supplier_Code (key), Supplier_Name, Product_Name, Product_ID (fk), Supplier_Phone, Supplier_Email)
Sample Data:

Supplier_Code	Supplier_Name	Product_ID	Supplier _Phone	Supplier_Email
S101	Alan's Farm	PD101, PD102	607-222-2222	Alanfarm@gmail.com
S102	Meet Fresh	PD103	607-222-2223	contact@meetfresh.com
S103	Sunshine Farmer	PD104	607-222-2224	sunshinefarmer@live.com
S104	Tompkins Farmacy	PD104	607-222-2225	customer@tompkinsfarmacy.com
S105	Betty's Farm	PD105	607-222-2226	betty@gmail.com

Key: Supplier_Code
FD1: Supplier_Code -> Supplier_Name, Supplier _Phone, Supplier_Email
FD2: Supplier_Code, Product_ID ->Supplier_Name, Supplier _Phone, Supplier_Email
1NF: Meets the definition of a relation
2NF: Partial Key dependencies exists: Supplier_Code, Product_ID ->Supplier_Name, Supplier _Phone, Supplier_Email and Supplier_Code -> Supplier_Name, Supplier _Phone, Supplier_Email
Solution: split Suppliers relation into two new relations named Suppliers and Products_Suppliers:

Suppliers(Supplier_Code (key), Supplier_Name, Supplier_Phone, Supplier_Email)
Key: Supplier_Code
FD1: Supplier_Code -> Supplier_Name, Supplier _Phone, Supplier_Email
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies
BCNF: All determinants are candidate keys

Products_Suppliers(Products_Suppliers_ID (key), Products_ID (fk), Supplier_Code (fk))
Key: Products_Suppliers_ID
FD1: Products_Suppliers_ID -> Products_ID, Supplier_Cod
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

2) **Products**(Products_ID (key), Product_Name, Supplier_Code (fk), Product_Price)
Sample Data:

Product_ID	Product_Name	Product_Price
PD101	Orange Juice	3.99
PD102	Apple Juice	3.99
PD103	Peach Juice	4.99
PD104	Kiwi Juice	4.99
PD105	Mango Juice	5.99

Key: Product_ID
FD1: Product_ID -> Product_Name, Product_Price
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies
BCNF: All determinants are candidate keys

3) **Customers**(Customers_ID (key), First_Name, Last_Name, Customers_Phone, Customers_Email)
Sample Data:

Customer_ID	First_Name	Last_Name	Customer_Phone	Customer_Email
C101	Elia	Fawcett	201-222-8888	ef88@gmail.com
C102	Ishwarya	Roberts	201-222-3333	ishwarya@deloitte.com
C103	Frederic	Fawcett	201-222-4424	frederic.fawcett@gmail.com
C104	Goldie	Montand	201-222-9999	2229999@qq.com
C105	Dheeraj	Alexander	201-222-4367	peace_alex@baruchmail.cuny.edu

Key: Customers_ID
FD1: Customers_ID -> First_Name, Last_Name, Customers_Phone, Customers_Email
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies
BCNF: All determinants are candidate keys

4) **Addresses**(Address_ID (key), Customer_ID, Line_1, Line_2, City, Zip, State)
Sample_Data

Address_ID	Customer_ID	Line_1	Line_2	City	Zip	State
AD10101	C101	46 Easton Ave		Garfield	07026	NJ
AD10102	C101	12 Fortis Blvd	Apt. 2A	Jersey City	07310	NJ

AD10201	C102	70 Willard St		Ithaca	14850	NY
AD10301	C103	73 Holly Terrace		Albany	12084	NJ
AD10401	C104	2 Lincoln Place		Stamford	06831	CT
AD10501	C105	215 Elm Ave		New York	10010	NY

Key: Address_ID
FD1: Address_ID -> Line_1, Line_2, City, Zip, State
FD2: Address_ID, Customer_ID -> Line_1, Line_2, City, Zip, State
1NF: Meets the definition of a relation
2NF: Partial Key dependencies exists: Address_ID, Customer_ID -> Line_1, Line_2, City, Zip, State and Address_ID -> Line_1, Line_2, City, Zip, State
Solution: split Addresses relation into two new relations named Addresses and Customers_Addresses:

Addresses(Address_ID (key), Line_1, Line_2, City, Zip, State)
Key: Address_ID
FD1: Address_ID -> Line_1, Line_2, City, Zip, State
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies
BCNF: All determinants are candidate keys

Customers_Addresses(Customers_Address_ID(key), Customer_ID(fk), Address_ID(fk))
Key: Customers_Address_ID
FD1: Customers_Address_ID -> Customer_ID, Address_ID
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies
BCNF: All determinants are candidate keys
5) **Customer_Orders**(Order_ID (key), Customer_ID (fk), PaymentMethod_ID (fk), Date_Order_Placed, Date_Order_Paid, Order_Price, Date_Reported, Delivery_Status, Product_ID, Quantity)
Sample Data

Order_ID	Customer_ID	PaymentMethod_ID	Date_Order_Placed	Date_Order_Paid	Order_Price	Date_Reported	Delivery_Status	Product_ID	Quantity
O101	C101	PM101	June 23, 2019	June 26, 2019	16.96	June 23, 2019	In transit	PD101	3
O101	C101	PM101	June 23, 2019	June 26, 2019	16.96	June 23, 2019	In transit	PD104	1
O102	C102	PM102	July 2, 2019	July 10, 2019	15.96	July 2, 2019	In transit	PD101	4
O103	C101	PM103	July 14, 2019	July 16, 2019	19.95	July 14, 2019	Delivered	PD103	5
O104	C103	PM104	Aug 19, 2019	Aug 19, 2019	19.95	Aug 19, 2019	Returned	PD102	5
O105	C104	PM105	Aug 20, 2019	Aug 21, 2019	29.95	Aug 20, 2019	In transit	PD105	5

Key: Order_ID
FD1: Order_ID->Customer_ID, PaymentMethod_ID, Date_Order_Placed, Date_Order_Paid, Order_Price, Date_Reported, Delivery_Status, Product_ID, Quantity
FD2: Order_ID, Delivery_Status -> Date_Reported
FD3: Order_ID, Product_ID -> Quantity
1NF: Meets the definition of a relation
2NF: Partial Key dependencies exists
Solution: split Customer_Orders relation into three relations named Customer_Orders and Customers_Orders_Product, and Orders_Delivery:

Customer_Orders(Order_ID (key), Customer_ID (fk), PaymentMethod_ID, Date_Order_Placed, Date_Order_Paid, Order_Price)
Key: Order_ID
FD1: Order_ID->Customer_ID, PaymentMethod_ID, Date_Order_Placed, Date_Order_Paid, Order_Price
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies
BCNF: All determinants are candidate keys

Customer_Orders_Product(Order_ID (key1), Product_ID (key2), Quantity)
Key: Order_ID, Product_ID
FD1: Order_ID, Product_ID -> Quantity
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies
BCNF: All determinants are candidate keys

Orders_Delivery(Order_ID (key1), Date_Reported, Delivery_Status (key2))
Key: Order_ID, Delivery_Status
FD1: Order_ID, Delivery_Status-> Date_Reported
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies
BCNF: All determinants are candidate keys

6) **Orders_Payment_Method**(PaymentMethod_ID (key), Order_ID(fk), Payment_Method_Code, Card_Number)
Sample Data

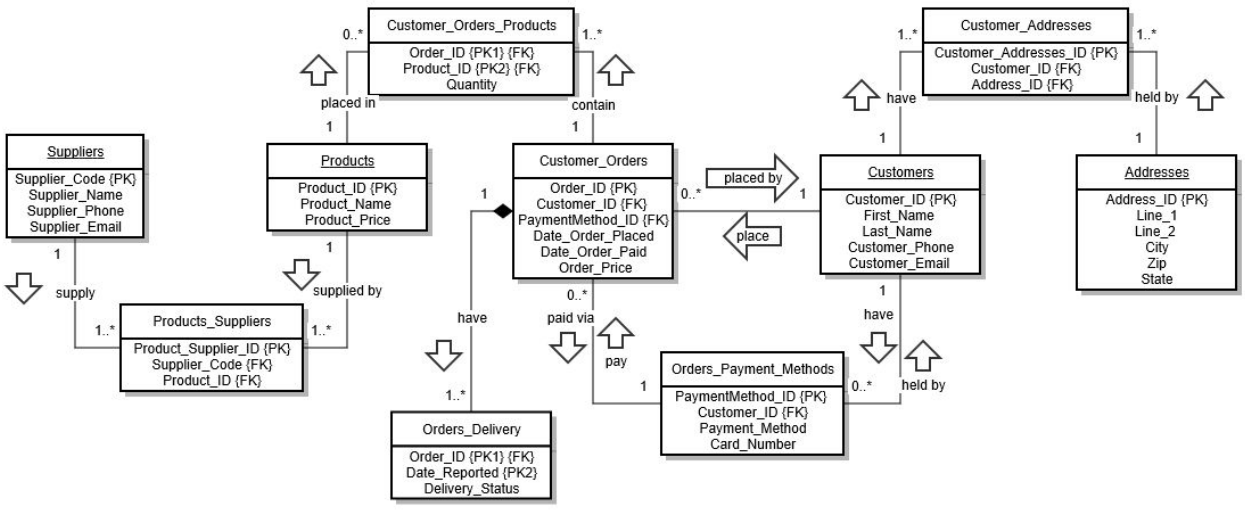
PaymentMethod_ID	Order_ID	Payment_Method_Code	Card_Number
PM101	O101	Visa	1111-2222-3333-4444
PM102	O102	Master	1111-2222-8934-5696
PM103	O103	Discover	1111-2222-1213-9823

PM104	O104	AmericanExpress	1111-2222-7812-9347
PM105	O105	Visa	1111-2222-2939-7365

Key: PaymentMethod_ID
FD1: PaymentMethod_ID ->Order_ID, Payment_Method_Code, Card_Number
1NF: Meets the definition of a relation
2NF: No partial Key dependencies
3NF: No Transitive dependencies
BCNF: All determinants are candidate keys

Final Set of Relations

Suppliers(Supplier_Code{PK}, Supplier_Name, Supplier_Phone, Supplier_Email)
Products_Suppliers(Products_Suppliers_ID, Products_ID {PK1}, Supplier_Code{PK2})
Products(Products_ID {PK}, Product_Name, Product_Price)
Customers(Customers_ID {PK}, First_Name, Last_Name, Customers_Phone, Customers_Email)
Customers_Addresses(Customer_ID{PK1}, Address_ID {PK2})
Addresses(Address_ID {PK}, Line_1, Line_2, City, Zip, State)
Customer_Orders(Order_ID {PK}, Customer_ID {FK}, PaymentMethod_ID, Date_Order_Placed, Date_Order_Paid, Order_Price)
Customer_Orders_Product(OrderID {PK1}, Product_ID {PK2}, Quantity)
Orders_Payment_Method(PaymentMethod_ID {PK}, Order_ID{FK}, Payment_Method_Code, Card_Number)
Orders_Delivery(Order_ID {PK1}, Date_Reported, Delivery_Status {PK2})



VI. Thoughts and Reflections on this project

We went through a lot of changes deciding the business scenario and the scale of the database intended. The final deliverable was largely different from our original submission for the mid-term deliverable. After discussing feedback with Professor Kline on our mid-term performance, we acquired a lot deeper understanding on the logic and design that didn't make sense, and during exhaustive discussion, we agreed on a model that actually came out before its ER model.

We talked about ensuring this relationship model fitting all the normalization requirements up to 3NF numerous times, and we are proud of our achievements within a limited time. However, we encountered some difficulties recreating the evolution from the ER model to our current relational model. We "assumed" a simple ER model that doesn't include our current intersection relations, thinking that would exactly be the ancient stage of what we've achieved today. However during normalization, we realized some relations in the ER model we "simply imagined" either had duplicate attributes, unnecessary attributes, or had relationships with other relations that cannot derive into our relational model via normalization. If we were to redo this project, we should stay true to the process which produces ER model first before a relational model, rather than "hopping through" stages.

Another challenge we faced was that the multiplicity of each relationship was discussed multiple times, however each time some new interpretation has derived from the last. Looking back we should have documented with clear details on our final agreement on these design details, so we won't still get confused sometimes in the last days of the project.

Overall we've learned a precious lesson from this project. We now have solid understanding on what's a bad design, what details it takes to maximize a good design, how to collaborate as a team as if each team member is positioned at various real life roles, such as database designer responsible for different model stages, project manager, and SQL programmer. Although we aren't yet experienced enough to create a flawless model, we thank this experience full of trials and errors.