

DATABASE DEVELOPMENT PROJECT
FOR



COLLECTIVE LIFESTYLE

8906 Reseda Blvd.
Northridge, CA 91324

PRESENTED BY
TEAM SUPREME CONSULTANTS

Aguirre, Joshua
Sharma, Pranab
Tovar, Miguel
Veran, Jared

EXECUTIVE SUMMARY

Collective lifestyle is a modern company born of the vision of one owner, CJ, to provide a place that offers both online services and a showroom, for people to find the latest trends in clothing and accessories. The company has had positive responses from the community which it serves, however the systems that are currently in place are redundant, slow and require a strenuous upkeep on a daily basis. The business also is using multiple databases to store and reference vital business functions. Because Collective Lifestyle has both physical and digital sales, they require a central program and database that can successfully and efficiently track inventory, sales and orders. Currently, Collective Lifestyle has multiple systems that does not successfully communicate with each other, causing the owner to manually adjust inventory and update sales. What we propose to change within the business is to create a new system that allows Collective Lifestyle to sustain one central database that contains inventory, sales, orders that represents both physical and digital sales. While investigating the backbone of the business, we as consultants have found numerous ways in which the process of everyday business functions can be increased by implementing a new method that can simultaneously collaborate multiple systems into one central database.

BUSINESS BACKGROUND

Collective Lifestyle LA

8906 Reseda Blvd.

Northridge, CA 91324

Contact info

Store Phone: (818) 280-6930

<http://collectivelifestyle.com>

CJ Berina founded Collective Lifestyle in May of 2014. When he started this store, he wanted to pursue the absolute most unique styles of fashion, accessories, and art. What is found most attractive by the eyes is not always that which is most colorful, but that which is effective in conveying simplicity; and Berina exemplifies just that in his business, both online and in-store as well.

Collective Lifestyle is a modern and contemporary clothing store located in the heart of the San Fernando Valley. This is a family owned store which aims to represent family morals. They seek to create a personal and family-style relationship with their customers, because by knowing their customer's interests and styles, they are better able to assist them in finding the proper apparel to buy. This is a clothing store which encourages customers to explore themselves in a profound manner to better express their inner-selves in the clothing they wear. This store carries a range of clothes for all seasons, for both men and women, and accessories which are as just unique as their clothing lines. Along with the clothing line, Collective Lifestyle promotes art paintings and drawings.

PROBLEM

The pre-existing system that Collective Lifestyle currently uses is outdated. Change is needed in order to progress growth, which is hindered with an outdated system. The growth that has occurred in recent months has led the company to conduct business in different forms of matter. The current databases are not on par with current business practices. In other words, we need to create new databases and form new relations. The three new databases that we need to develop are Customer's Order Receipt Data, Product Supplier Information, Product Inventory and Pricing. It is important to have current data intertwining cohesively so the migration can be easy toward the new system.

PROJECT SCOPE AND OBJECTIVES

Team Supreme is ready to develop and deploy a new database for Collective Lifestyle. Our project will cover the ordering process from the customer to the supplier. Every form of data is information, to the smallest detail. Collective Lifestyle is consider a third party seller, so the information that a product takes to reach A to Z is grand in importance

We will develop the system with the following objectives in mind:

- (1) System will have availability by have little to no bugs.
- (2) The system will be built with quality that is on par with our CSUN education
- (3) Team Supreme will provide on the call support to Collective lifestyle.
- (4) We will constantly inform our client with cohesive updates.

BUSINESS RULES

1. Each **ORDER** must have at least one **PRODUCT**. Each **PRODUCT** may be used to fill many **ORDER(s)**.

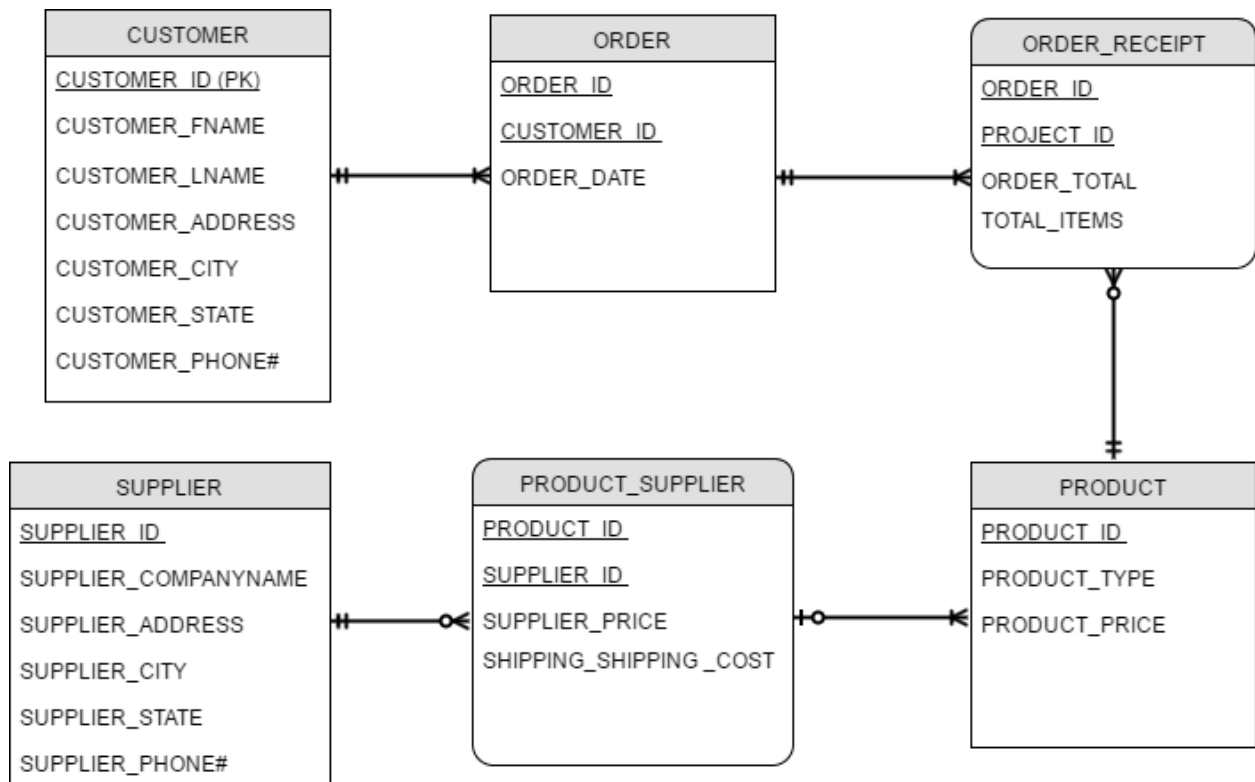
- | |
|---|
| <ol style="list-style-type: none">1. ORDER and PRODUCT is a M - M relationship because you must have at least one PRODUCT for each ORDER and a ORDER must have a PRODUCT.2. The cardinality for ORDER is mandatory one.3. The cardinality for PRODUCT is mandatory many. |
|---|

2. Each **CUSTOMER** may have one or more **ORDER(s)**. However, each **ORDER** must belong to exactly one **CUSTOMER**.

- | |
|--|
| <ol style="list-style-type: none">1. CUSTOMER and ORDER is a 1-M relationship because one ORDER must have one CUSTOMER and one CUSTOMER may have multiple ORDER(s).2. The cardinality for ORDER is mandatory many.3. The cardinality for CUSTOMER is mandatory one. |
|--|

DATA MODELING DIAGRAMS

ERD



RELATIONAL MODEL DIAGRAM

CUSTOMER

<u>CUSTOMER_ID</u>	CUSTOMER_FNAME	CUSTOMER_LNAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_STATE	CUSTOMER_PHONE
--------------------	----------------	----------------	------------------	---------------	----------------	----------------

ORDER

<u>ORDER_ID</u>	<u>CUSTOMER_ID</u>	ORDER_DATE
-----------------	--------------------	------------

ORDER_RECEIPT

<u>ORDER_ID</u>	<u>PROJECT_ID</u>	ORDER_TOTAL	TOTAL_ITEMS
-----------------	-------------------	-------------	-------------

PRODUCT

<u>PRODUCT_ID</u>	PRODUCT_TYPE	PRODUCT_PRICE
-------------------	--------------	---------------

PRODUCT SUPPLIER

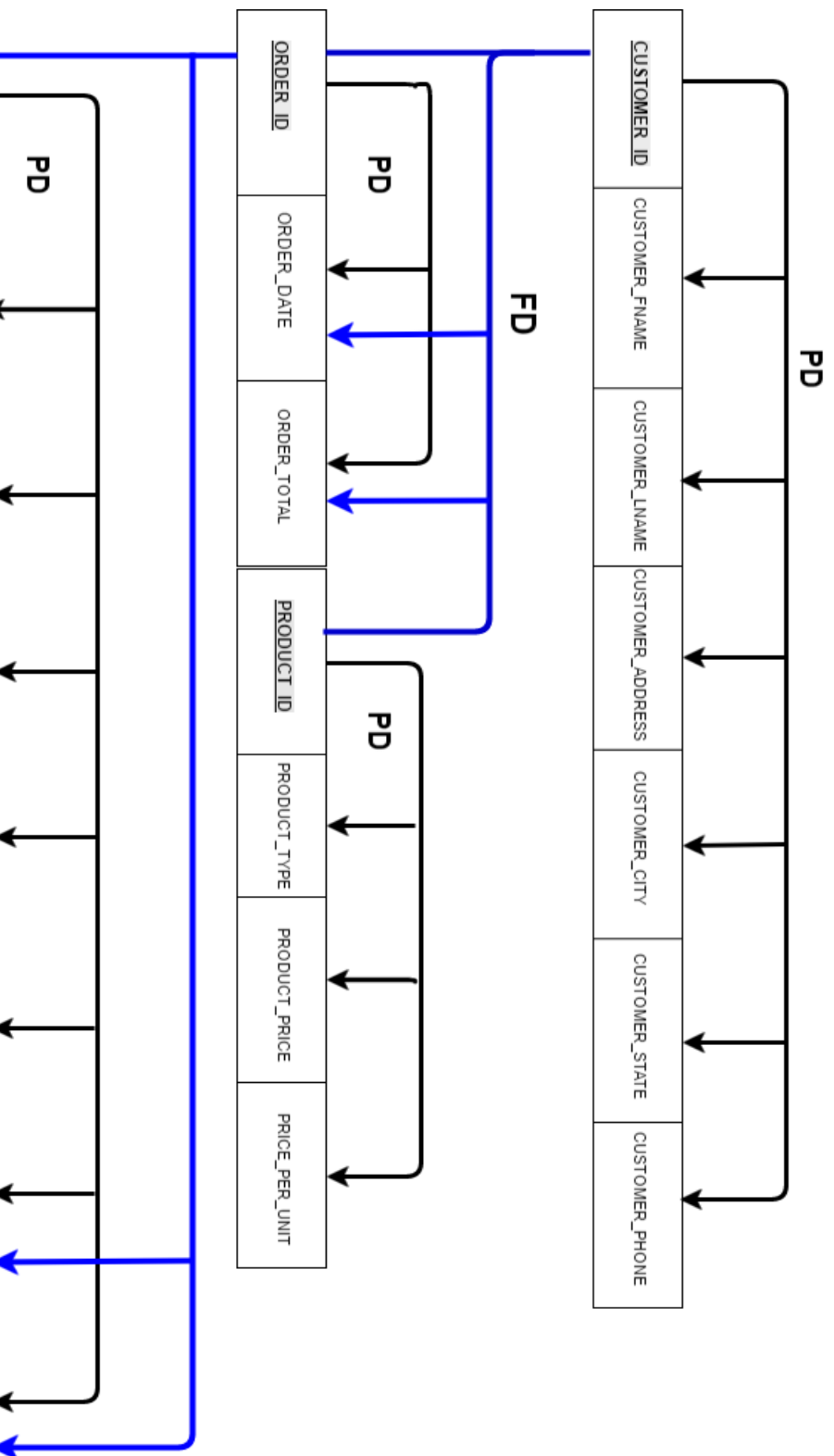
<u>PRODUCT_ID</u>	<u>SUPPLIER_ID</u>	SUPPLIER_PRICE	SHIPPING_COST
-------------------	--------------------	----------------	---------------

SUPPLIER

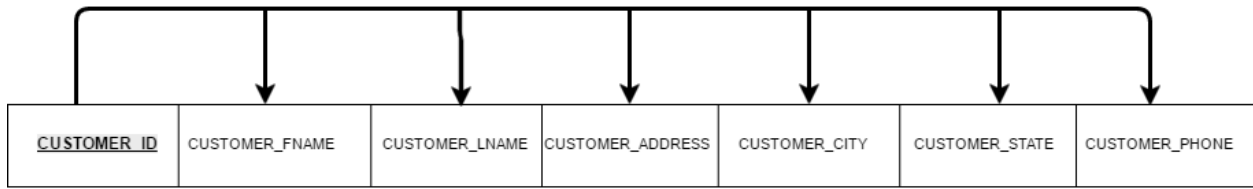
<u>SUPPLIER_ID</u>	SUPPLIER_COMPANYNAME	SUPPLIER_ADDRESS	SUPPLIER_CITY	SUPPLIER_STATE	SUPPLIER_PHONE#
--------------------	----------------------	------------------	---------------	----------------	-----------------

NORMALIZATION CHARTS

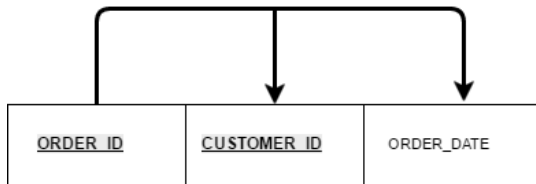
1NF



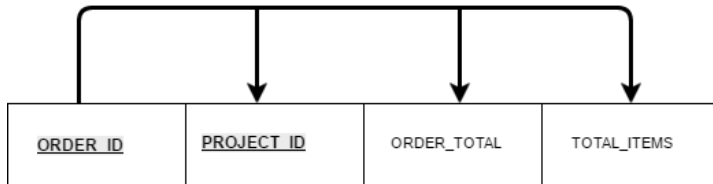
2NF



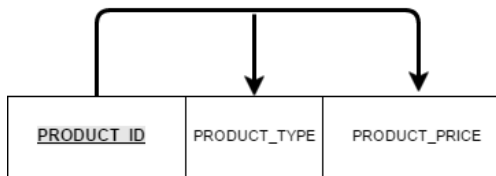
CUSTOMER



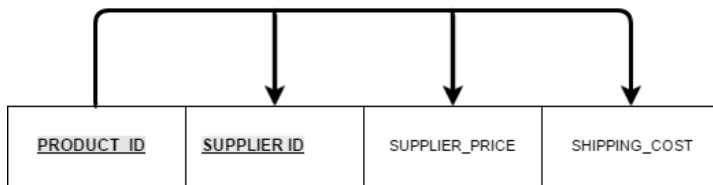
ORDER



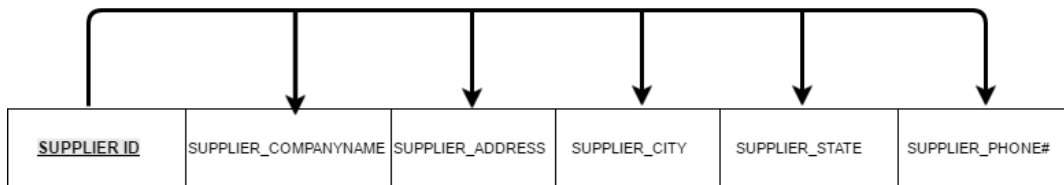
ORDER_RECEIPT



PRODUCT

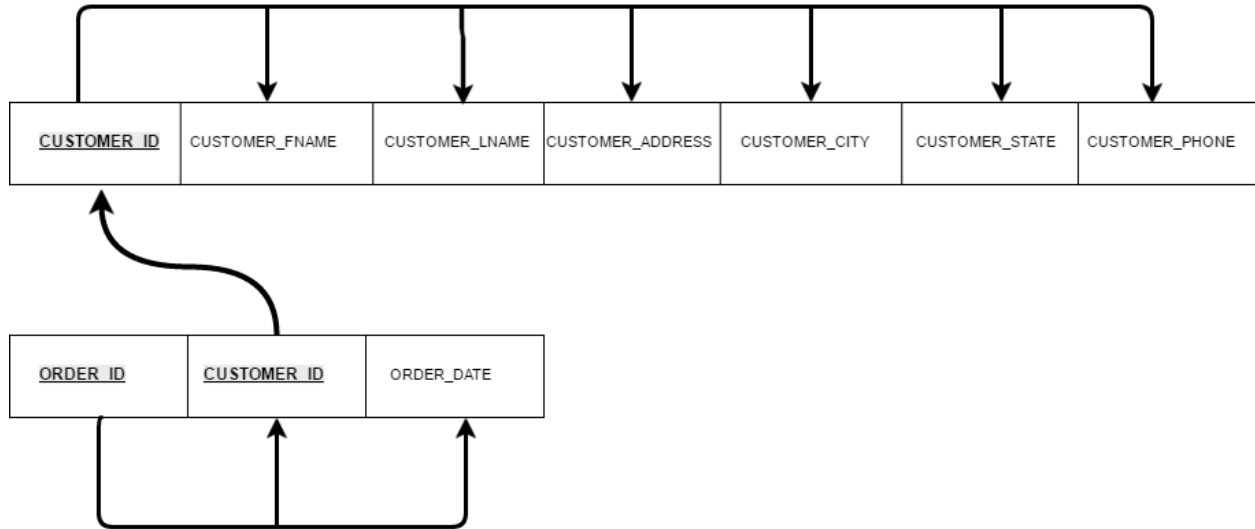


PRODUCT_SUPPLIER



SUPPLIER

3NF



QUERY DESIGNS

CUSTOMER TOTALS

QUERY 1

Description: The following SQL code displays the totals for each customers that order are over \$100. The code establishes a relationship with table ORDER_RECEIPT, ORDER and CUSTOMER.

SQL CODE

```
SELECT CUSTOMER.CUSTOMER_ID, LNAME, ORDER_TOTAL
FROM ORDER_RECEIPT, [ORDER], CUSTOMER
WHERE ORDER_RECEIPT.ORDER_ID = ORDER.ORDER_ID
AND ORDER.CUSTOMER_ID = CUSTOMER.CUSTOMER_ID
AND ORDER_TOTAL >= 100
ORDER BY CUSTOMER.CUSTOMER_ID ;
```

SQL OUTPUT

CUSTOMER_	LNAME	ORDER_TOTAL
1	Gruber	\$217.60
3	Gantt	\$208.85
4	Snow	\$185.42
5	Mata	\$157.74
6	Hancock	\$163.73
8	Arnold	\$181.17
9	Ainsworth	\$212.52
10	Hanley	\$276.85
11	Slaugh	\$484.36
12	Parente	\$185.33

PRODUCT TYPE TOTALS

QUERY 2

Description: This code below shows the total price for type of product for supplier that are located in Mississippi, Texas and Ohio.

SQL

```
SELECT PRODUCT_TYPE, SUM(PRICE) AS TOTALSUM
FROM [PRODUCT], [PRODUCT_SUPPLIER], [SUPPLIER]
WHERE PRODUCT.PRODUCT_ID = PRODUCT_SUPPLIER.PRODUCT_ID
AND PRODUCT_SUPPLIER.SUPPLIER_ID = SUPPLIER.SUPPLIER_ID
AND STATE IN ('MS', 'TX', 'OH')
GROUP BY PRODUCT.PRODUCT_TYPE;
```

SQL OUTPUT

PRODUCT_TYPE	TOTALSUM
pants	\$515.38
shirt	\$313.17
shoe	\$519.55

PROFIT TOTALS

QUERY 3

Description: The follow code shows the total profit from each product that was sold and on the side shows total cost and sale price.

SQL
SELECT PRODUCT.PRODUCT_ID, (SUPPLIER_PRICE + SHIPPING_COST) AS TOTALCOST, PRODUCT.PRICE AS SALE_PRICE, (PRODUCT.PRICE - (SUPPLIER_PRICE + SHIPPING_COST)) AS PROFIT FROM [PRODUCT_SUPPLIER], [PRODUCT] WHERE PRODUCT.PRODUCT_ID = PRODUCT_SUPPLIER.PRODUCT_ID;

OUTPUT ON NEXT PAGE.....

SQL OUTPUT

PRODUCT_ID	TOTALCOST	SALE_PRICE	PROFIT
5891	\$76.55	\$77.00	\$0.45
8835	\$40.71	\$45.99	\$5.28
9827	\$35.31	\$40.77	\$5.46
38037	\$107.21	\$112.07	\$4.86
45340	\$99.56	\$105.53	\$5.97
56380	\$54.58	\$58.20	\$3.62
56450	\$83.23	\$86.71	\$3.48
56460	\$61.84	\$63.94	\$2.10
60399	\$22.62	\$34.54	\$11.92
72336	\$24.30	\$28.97	\$4.67
75262	\$85.09	\$86.64	\$1.55
75291	\$27.13	\$29.33	\$2.20
86591	\$70.67	\$71.00	\$0.33
89320	\$57.59	\$60.19	\$2.60
90859	\$72.83	\$76.46	\$3.63
91266	\$42.55	\$48.32	\$5.77
95936	\$84.00	\$87.74	\$3.74
97436	\$14.67	\$18.26	\$3.59
245770	\$104.48	\$108.71	\$4.23
246560	\$78.39	\$81.58	\$3.19
254423	\$30.56	\$31.32	\$0.76
345340	\$106.30	\$111.16	\$4.86
345450	\$23.66	\$27.19	\$3.53
345560	\$41.24	\$44.22	\$2.98
435646	\$40.18	\$43.63	\$3.45
456540	\$59.58	\$64.18	\$4.60
465640	\$52.69	\$93.56	\$40.87
493176	\$36.15	\$56.97	\$20.82
515393	\$63.93	\$93.36	\$29.43
542520	\$63.09	\$66.24	\$3.15
564460	\$22.01	\$24.69	\$2.68
766129	\$14.57	\$20.87	\$6.30

CUSTOMER TOTALS

QUERY 4

Description: The following codes show the number of orders that were place that day with alongside with total items and total sales for that day.

SQL

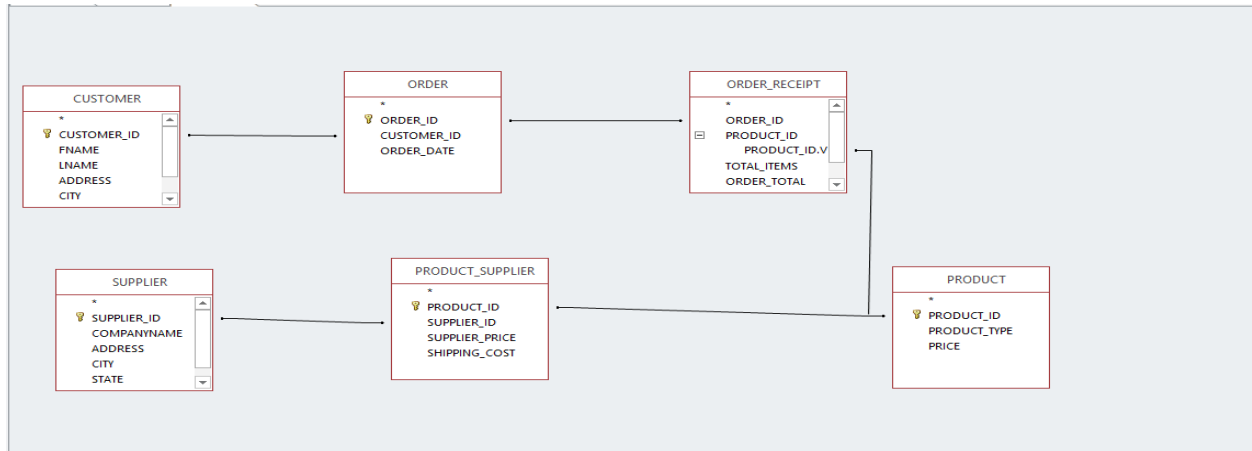
```
SELECT ORDER_DATE, COUNT(ORDER_DATE) AS NUMOFORDERS,  
SUM(TOTAL_ITEMS) AS TOTALITEMS, SUM (ORDER_TOTAL) AS DAYTOTAL  
FROM [ORDER], ORDER_RECEIPT  
WHERE ORDER.ORDER_ID = ORDER_RECEIPT.ORDER_ID  
GROUP BY ORDER_DATE;
```

SQL OUTPUT

ORDER_DATE	NUMOFORDERS	TOTALITEMS	DAYTOTAL
4/2/2017	1	2	\$45.56
4/3/2017	3	8	\$401.13
4/4/2017	3	7	\$412.15
4/5/2017	3	13	\$882.21
4/6/2017	2	3	\$283.84
4/7/2017	1	1	\$56.97
4/8/2017	1	5	\$276.85

USER MANUAL

Overview



1. **First**, employee must input customer data into the database
2. **Second**, employee must place order to order database and order_receipt database.
3. **Third**, Product type must be recorded alongside with its attributes.
4. **Fourth**, Product must be match with supplier in database
 - a. Supplier information is incorporated to database as reference.

Invoice



Collective Lifestyle

INVOICE

Date: \$Date
ORDER # \$ORDER_ID

To \$CUSTOMER_FNAME
\$CUSTOMER_LNAME
\$CUSTOMER_ADDRESS
\$CUSTOMER_CITY,
\$CUSTOMER_STATE
\$CUSTOMER_PHONE#
Customer ID: \$CUSTOMER_ID

Salesperson	Job	Payment Terms	Due Date
		Due on receipt	

Qty	Description	Unit Price	Line Total
Subtotal			
Sales Tax			
Total			

Make all checks payable to Collective Lifestyle

Collective Lifestyle 8906 Reseda Blvd. Northridge, CA 91324 Phone: 818-280-6930

A customer and or supplier must have an invoice depending on the type of purchase.

When an invoice is created the user must make sure the type of purchase is specified. An in store invoice must involve the CUSTOMER_ID. Invoices can be manually generated or you can use the database to export into the variables with the macro that will be provided.

Other Recommendations

The purpose of this system is to organize and store data efficiently. Data can be analyzed and improve customer relations and profits. Thus overtime implementing more complex queries may be necessary. Especially when entering and analyzing significantly larger data stores. Thus for future use more fields and nominalization may be necessary in order to streamline the process. A database update will be necessary in the future until then we can our current databases exceeds the expectations that this business needs. If Collective Lifestyle wants a more in depth understanding of their new database, we can recommend that he should undertake training or classes. This can enable him to achieve even more complex queries and to generate complex reports for both employees

APPENDIX

Customer Report

CUSTOMER				Monday, May 8, 2017 12:31:17 AM
CUSTOMER_ID	FNAME	LNAME	ADDRESS	
1	Thomas	Gruber	889 Simpson Avenue	
2	Christopher	Price	3387 Coolidge Street	
3	Katherine	Gantt	3140 Eagles Nest Drive	
4	Nicole	Snow	511 Oakmound Drive	
5	Deborah	Mata	2895 Highland Drive	
6	Matthew	Hancock	748 East Avenue	
7	Nguyễn	Mộng Vui	3196 Holden Street	
8	Orsós	Arnold	1841 Elm Drive	
9	Modesto	Ainsworth	4181 Francis Mine	
10	Ronald	Hanley	3098 Denver Avenue	
11	Meredith	Slaugh	1717 Coplin Avenue	
12	Raul	Parente	2236 Southside Lane	
13	Rodney	Perez	1568 Elk Street	
14	William	Harris	558 Bicetown Road	
15	John	Watkins	1900 Shinn Avenue	
15				

Customers can be added at any time and a point of sale system can allow the salesperson to populate this table. It is important that the sales person fill out the fields to maintain integrity.

Order Report

ORDER		
ORDER_ID	CUSTOMER_ID	ORDER_DATE
1	6	4/7/2017
2	1	4/6/2017
3	3	4/4/2017
4	15	4/3/2017
5	10	4/8/2017
6	9	4/5/2017
7	11	4/5/2017
8	8	4/3/2017
9	5	4/4/2017
10	2	4/7/2017
11	7	4/2/2017
12	14	4/6/2017
13	12	4/5/2017
14	4	4/3/2017
15	13	4/4/2017
15		

Page 1 of 1

An employee can be added only by the owner of the store as he would be the only one with write access to the database.

Order_Receipt Report

ORDER_RECEIPT				Monday, May 8, 2017
				12:32:13 AM
ORDER_ID	PRODUCT_ID	TOTAL_ITEMS	ORDER_TOTAL	
	5891, 8835, 9827	3	\$163.73	
2	38037, 45340	2	\$217.60	
3	56380, 56450, 56460	3	\$208.85	
4	60399	1	\$34.54	
5	72336, 75262, 75291, 86591, 89320	5	\$276.85	
6	90859, 91266, 95936	3	\$212.52	
7	95936, 97436, 245770, 246560, 254423, 345325, 345340	7	\$484.36	
8	345450, 345560, 435347, 435646	4	\$181.17	
9	456450, 456540	2	\$157.74	
10	493176	1	\$56.97	
11	564460, 766129	2	\$45.56	
12	542520	1	\$66.24	
13	456540, 465640, 493176	3	\$185.33	
14	245770, 435646, 456450	3	\$185.42	
15	56450, 75291	2	\$45.56	
			\$2,522.44	

The order receipt contains information about the purchase or order. Because this is automatically generated, no users should modify this.

Supply Chain Report

PRODUCT_SUPPLIER				Monday, May 8, 2017	
				12:33:35 AM	
PRODUCT_ID	SUPPLIER_ID	SUPPLIER_PRICE	SHIPPING_COST		
5891	1	\$73.34	\$3.21		
8835	3	\$37.34	\$3.37		
9827	1	\$32.40	\$2.91		
38037	3	\$103.34	\$3.87		
45340	5	\$97.32	\$2.24		
56380	5	\$52.53	\$2.05		
56450	3	\$82.28	\$0.95		
56460	6	\$58.53	\$3.31		
60399	2	\$21.18	\$1.44		
72336	4	\$20.86	\$3.44		
75262	6	\$81.73	\$3.36		
75291	4	\$24.67	\$2.46		
86591	2	\$67.34	\$3.33		
89320	1	\$53.56	\$4.03		
90859	6	\$68.80	\$4.03		
91266	4	\$39.51	\$3.04		
95936	3	\$81.43	\$2.57		
97436	2	\$12.63	\$2.04		
245770	5	\$99.83	\$4.65		
246560	5	\$75.24	\$3.15		
456540	5	\$57.32	\$2.26		
435646	3	\$37.34	\$2.84		
766129	3	\$13.50	\$1.07		
564460	1	\$20.42	\$1.59		
542520	1	\$58.64	\$4.45		
515393	2	\$61.06	\$2.87		
345340	2	\$102.43	\$3.87		
254423	4	\$27.92	\$2.64		
493176	2	\$32.76	\$3.39		
345560	4	\$37.34	\$3.90		
345450	3	\$21.43	\$2.23		
PRODUCT_ID	SUPPLIER_ID	SUPPLIER_PRICE	SHIPPING_COST		
465640	1	\$50.23	\$2.46		
				\$93.06	

The supply chain information can be found here. This will be useful in terms of restocking items and for ordering special orders.