Online Sales Management System

for

LACOSTE

Guided by: Prof. Sreeram Chavali

Nidhee Dumbhre | DBMS | 9 February 2022

Table of Contents

INTRODUCTION

The advancement of technology is transforming our standards of living and lifestyle. Online shopping has been one of the prevalent features of modern technology. The number of online shoppers has been growing over the past few years. In 2021, there were 900 million more digital buyers than there were in 2020—a 4.4 percent year-over-year increase. Given the rising trend, the potential for an ecommerce store is huge. A deeper look into the e-shopping statistics can prove vital to the store owners for better understanding of the trends and thereby enhance the online shopping business. As a part of this project, we will create a database management system (DBMS) for online sales for Lacoste and leverage the concepts and features of relational DBMS to allow efficient administration and management of the data

OBJECTIVE

The goal of this study is to analyze online sales trend, sales based on demographics, demand of different sizes of products from the product line and reach of the brand amongst the diverse group of individuals. The study will navigate us through the patterns followed by different consumers, concentration of consumer market per locality/city and how the study of these patterns can be used to strengthen the marketing, advertising and even production strategies to maximize sales and eventually maximize the profit.

BUSINESS SCENARIO DESCRIPTION

My husband planned to purchase a sporty sweatshirt and started with a little research on which brand he wants to go with. After researching a few brands, he decided that he wanted to go with one of his favorite brands, Lacoste. Because of rising COVID cases, he was inclined to make an online purchase and hence started looking for sweatshirts on Lacoste website.

He started with browsing the Lacoste website and went through section of interest MEN CLOTHING SWEATSHIRTS. The section had 79 relevant items displayed. He looked through a few options and, in each option, he scanned different angled pictures provided. He liked one of the sweatshirts but found that his usual size wasn't available in stock. He then checked the "size guide" to ensure that he was looking for the right size. For the further search, he applied the "size" filter & "low to high" price filter to refine the results, there were 42 items available in his size. He decided to finalize on a Zip Pocket Hooded Sweatshirt after checking the reviews and customer posted images. The sweatshirt was from the winter sale collection and therefore on 50% off. He selected the size and added the sweatshirt into the bag and followed a pop-up "View bag" or "Continue Shopping". At the bottom of the checkout page there was a section "You may also be interested in" which

displayed different shirts, masks and sneakers that might go well with the sweatshirt. He looked through some of the suggested shirts and liked one of the shirts and added that to the bag.

On the checkout page, before making the payment, he was asked to enter his email address and create an account (and get subscribed to newsletter and exciting offers) or checkout as a guest. He created an account using his email id, updated shipping and billing address, and selected Ground Shipping (which was free) from 3 different shipping options. He then proceeded to the Payment Section and applied a 15% coupon code that he had found on the receipt of his last in-store purchase which was valid through June 2022. He selected the credit card option and reviewed his order to make the final purchase. Right after completing the order, he validated the confirmation email that he received for the order summary, delivery address and price information.

Dataset Creation

After making online purchases from Lacoste, a transaction log has been created, and the log has been extended to show sample transactions made by 13 other fictitious customers. The table with transaction logs was in de-normalized form and had redundant and repeated data. Moreover, there were insertion, deletion & updation anomalies. For example, deletion of a particular product in case a customer record is being deleted. By using normalization techniques, this dataset can be stored in different but related types of data in separate logical tables making the dataset free of any duplication and anomalies. We will start the process by first defining the process flow and then identifying entities involved in order to normalize the data.

PROCESS FLOWCHART

To visualize end-to-end workflow, we identified the roles of multiple groups involved in the business scenario and represented them in horizontal swimlanes. Then we captured each process step as a connection, interaction between same/different swimlanes (groups) to plot the complete process. Below are the details of groups involved:

Customers - Customers can have easy access to various products and their detailed description across the Lacoste website and can provide their views on overall shopping experience and the product purchased through ratings.

Website– Website is an interface between the customer and seller that provides a way to browse the product catalogue and perform online purchases.

Payment Gateway – A payment gateway provides a service to securely authorize payment made by customers for their purchases.

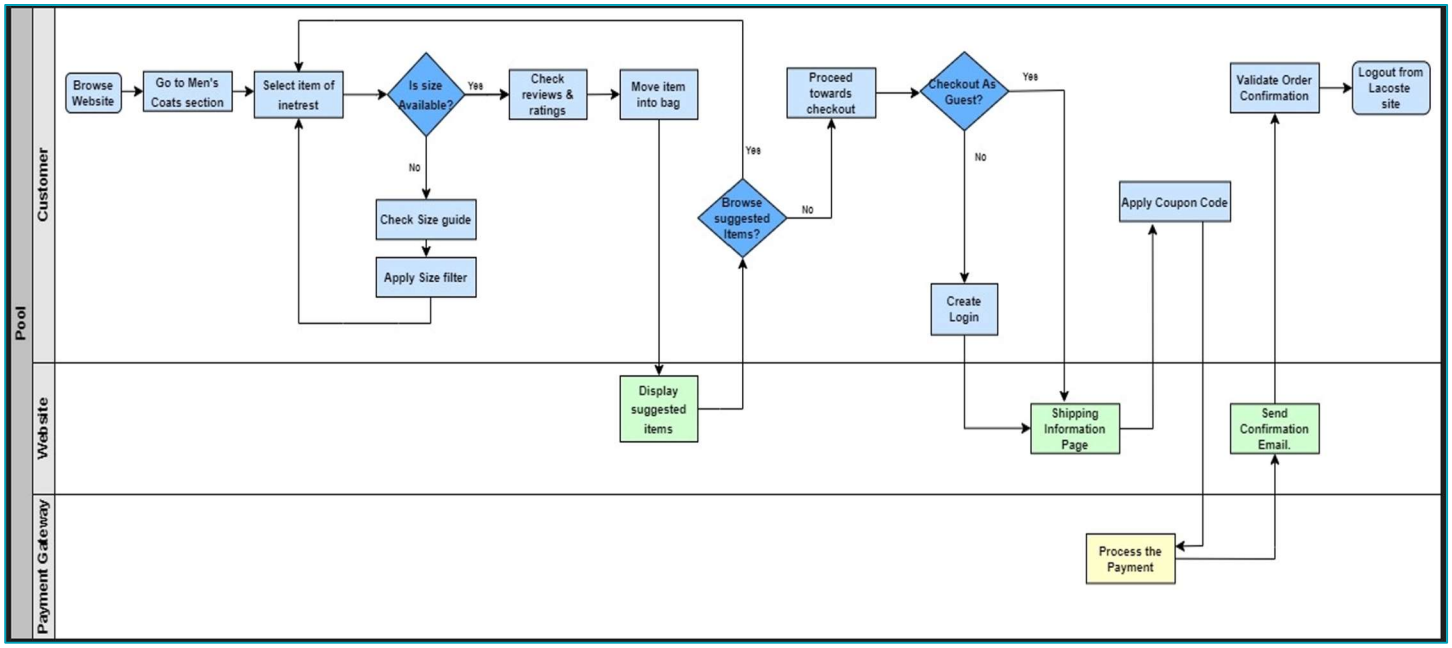


Fig. 1. Swim-lane diagram of the business case

ENTITY – RELATIONSHIP

After determining the groups through swim lane, we will now identify multiple entities involved in the business scenario and established their relationship based on the interactions and interdependencies. For each entity a table has been designated along with the information that needs to be stored in each table. The below UML model gives the details:

Note: **Transaction log** is deformalized table and is present there without any links only to show contents of original table

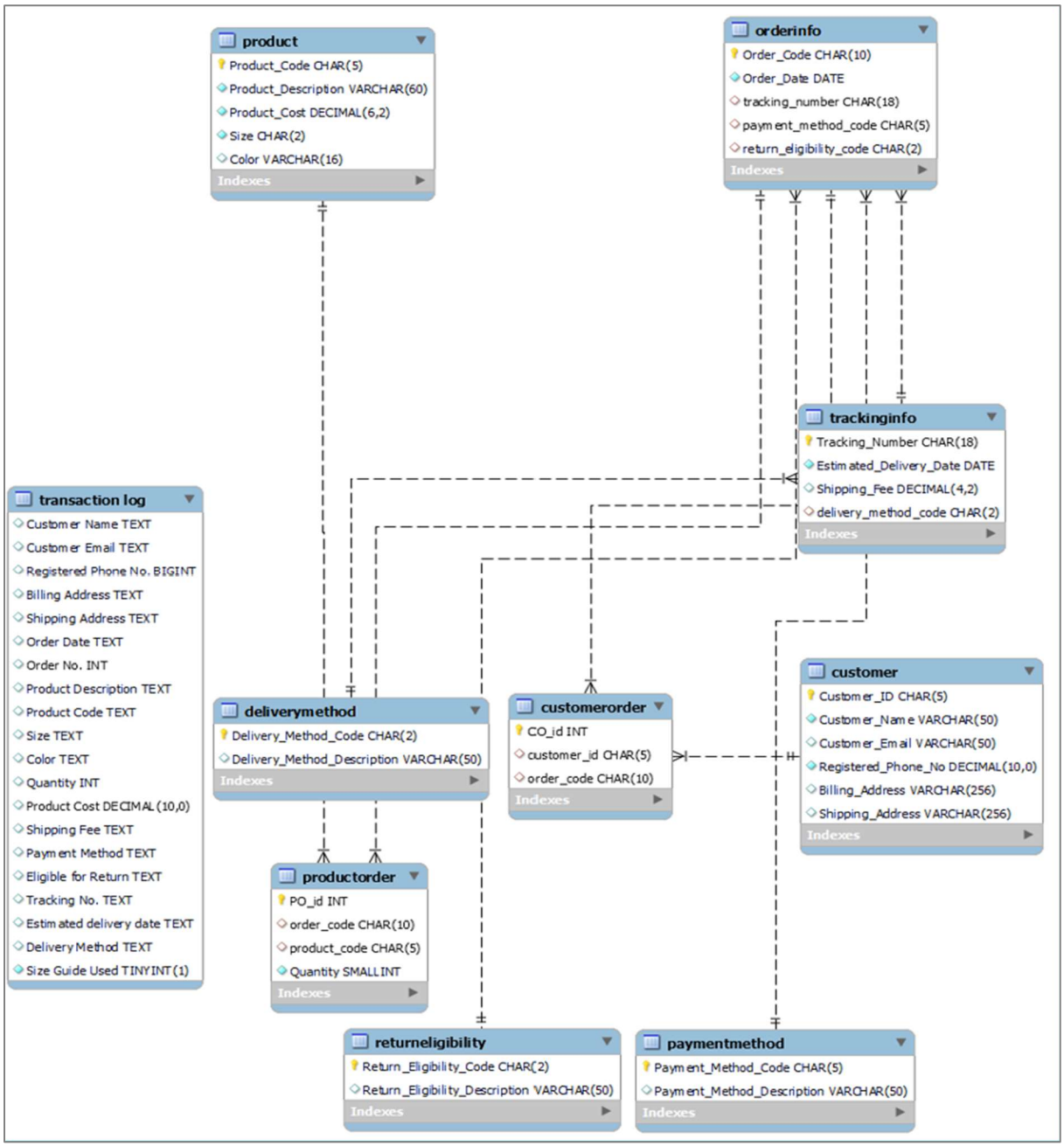


Fig. 2. Enhanced Entity Relationship (EER) Diagram

TABLES

LIST OF TABLES

Table: customer

Columns:

Customer_ID	char(5) PK
Customer_Name	varchar(50)
Customer_Email	varchar(50)
Registered_Phone_No	decimal(10,0)
Billing_Address	varchar(256)
Shipping_Address	varchar(256)

Table: product

Columns:

Product_Code	char(5) PK
Product_Description	varchar(60)
Product_Cost	decimal(6,2)
Size	char(2)
Color	varchar(16)

Table: paymentmethod

Columns:

Payment_Method_Code	char(5) PK
Payment_Method_Description	varchar(50)

Table: returneligibility

Columns:

Return_Eligibility_Code	char(2) PK
Return_Eligibility_Description	varchar(50)

Table: deliverymethod

Columns:

Delivery_Method_Code	char(2) PK
Delivery_Method_Description	varchar(50)

Table: trackinginfo

Columns:

Tracking_Number	char(18) PK
Estimated_Delivery_Date	date
Shipping_Fee	decimal(4,2)
delivery_method_code	char(2)

Table: orderinfo

Columns:

Order_Code	char(10) PK
Order_Date	date
tracking_number	char(18)
payment_method_code	char(5)
return_eligibility_code	char(2)

Table: productorder

Columns:

PO_id	int AI PK
order_code	char(10)
product_code	char(5)
Quantity	smallint

Table: customerorder

Columns:

CO_id	int AI PK
customer_id	char(5)
order_code	char(10)

TABLE CREATION SCRIPT

1. Starting with Creation of Schema - Create **SCHEMA** & upload "Transaction log "

```
## Create Schema
CREATE SCHEMA `Lacoste_Online_Sales`;
## Use created schema
USE `Lacoste_Online_Sales`;
```

2. Create Table – "**Customer**" with Primary key as Customer ID. Since we do not have a customer id, we will assign unique customer id to each customer.

```
CREATE TABLE Customer
(
Customer_ID CHAR(5),
Customer_Name VARCHAR(50) NOT NULL,
Customer_Email VARCHAR(50),
Registered_Phone_No DECIMAL(10) NOT NULL,
Billing_Address VARCHAR(256),
Shipping_Address VARCHAR (256),
CONSTRAINT PKCustomer_ID PRIMARY KEY (Customer_ID)
);
```

```
33 •  SELECT * FROM `Lacoste Online Sales`.`Customer`;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Customer_ID	Customer_Name	Customer_Email	Registered_Phone_No	Billing_Address	Shipping_Address
1	Shantanu Potdar	shaan.potdar@gmail.com	6667778989	10285 Parkwood Dr Apt 10 Cupertino	10285 Parkwood Dr Apt 10 Cupe
10	Jeff Gomez	jeff.g@mail.com	6780101210	500 El Camino Real Santa Clara	500 El Camino Real Santa Clara
11	Kylie John	kylie.j@mail.com	6664440555	Palm Court House No 456 Los Gatos	Palm Court House No 456 Los Ga
12	Sean Kernen	sean.k@mail.com	6780000123	1498 Wayfair Garden House No 704 Santa Monica	1498 Wayfair Garden House No
13	Rudi Josh	rudi.j@mail.com	6780901004	1040 Valleyfair Road Milpitas	1040 Valleyfair Road Milpitas
14	Will Richard	will.r@mail.com	6780901234	400 Park View Apt 23 Sunnyvale	400 Park View Apt 23 Sunnyvale
2	Pamela Anderson	pam.anderson@mail.com	6781230345	245 Park lane Rd Apt 4 Sunnyvale	Mindspace Omega-Building C San
3	Puja Rao	puja.r@mail.com	6668880999	Northseal Sq H No 101, Cupertino	Northseal Sq H No 101, Cupertin
4	Erica Blue	erica.b@mail.com	6663330444	Cresent Park Apt 202 Mountain View	North San Jose Tech Park Buildin
5	Ashley Turner	ashley.t@mail.com	6665550666	245 Park lane Rd Apt 21 Palo Alto	245 Park lane Rd Apt 21 Palo Alt
6	Ken Neil	ken.n@mail.com	6666660777	Northseal Sq H No 220 Cupertino	Northseal Sq H No 220 Cupertino

Customer 1 × Apply

It removed all the duplicate entries and in return we got 14 rows of non-duplicate data.

3. Create Table – "**Product**" using Product code as a Primary Key.

```
CREATE TABLE Product
(
Product_Code CHAR(5),
Product_Description VARCHAR(60) NOT NULL,
Product_Cost DECIMAL(6,2) NOT NULL,
Size CHAR(2) NOT NULL,
Color VARCHAR(16),
CONSTRAINT PKProduct_Code PRIMARY KEY (Product_Code),
CONSTRAINT CSize_Check CHECK (Size in ('XS','S','M', 'L', 'XL'))
```

```
);
```

```
71 •  SELECT * FROM `Lacoste Online Sales`.`Product`;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Product_Code	Product_Description	Product_Cost	Size	Color
00BBM	Unisex L.12.12 Adjustable Face Mask	45.00	L	Pink
01HBM	Mens Carnaby BL Leather Trainers	65.99	XL	White
02HBM	Mens Slim Fit Checkered Stretch Cotton shirt	143.99	L	Balck
08HBM	Mens Striped CashmereSweater	585.00	L	Yellow
0HBM	Mens Regular Fit Cotton Poplin Shirt	65.99	S	Black
14HBM	Boys Solid gabardine cap	35.00	XL	Black
18HBM	Mens Stretch Cotton Chino Pants	89.50	L	Blue
1HBM	Oval Bio Acetate National Geographic Sunglasses	128.99	S	Black
22HBM	Mens Striped Fleece Jogging Pants	76.99	XL	Blue
25HBM	Mens Cotton Blend Sock 3-Pack	83.96	L	White
29HBM	Womens SPORT Piqué Tennis Skirt	62.99	XS	White
2HBM	Womens Crew Neck Cotton T-shirt	49.50	XS	White

Product 2 | Apply

Note: * Product code is unique per product size & color

4. Create Table – "**Payment Method**" with Payment Method Code as Primary Key. We have assigned different and unique values for each payment method. Any updates or additions to the payment methods can now be performed independently without order or customer info dependencies.

```
CREATE TABLE PaymentMethod
(
Payment_Method_Code CHAR(5) NOT NULL,
Payment_Method_Description VARCHAR(50),
CONSTRAINT PKPayment_Method_Code PRIMARY KEY (Payment_Method_Code)
);
```

```
81 •  SELECT * FROM `Lacoste Online Sales`.`PaymentMethod`;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Payment_Method_Code	Payment_Method_Description
1	Credit Card
2	American Express
NULL	NULL

5. Create Table – "**Return Eligibility**" with Return Eligibility Code as Primary Key. We have assigned different and unique values for each return method.

```
CREATE TABLE ReturnEligibility
(
Return_Eligibility_Code CHAR(2) PRIMARY KEY NOT NULL,
Return_Eligibility_Description VARCHAR(50)
);
```

```
90 •    SELECT * FROM `Lacoste Online Sales`.`ReturnEligibility`;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Return_Eligibility_Code	Return_Eligibility_Description
1	Yes within 30 days of purchase
2	Not eleigible for return exchange
NULL	NULL

6. Create Table – "**Delivery Method**" with Delivery Method Code as Primary Key. We have assigned different and unique values for each delivery method.

```
CREATE TABLE DeliveryMethod
(
Delivery_Method_Code CHAR(2) PRIMARY KEY NOT NULL,
Delivery_Method_Description VARCHAR(50)
);
```

```
99 •    SELECT * FROM `Lacoste Online Sales`.`DeliveryMethod`;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Delivery_Method_Code	Delivery_Method_Description
1	At home Standard
2	At Office Standard
NULL	NULL

7. Create Table – "**Tracking Info**" with Tracking No. as Primary Key & Delivery Method Code as Foreign Key.

```
CREATE TABLE TrackingInfo
(
Tracking_Number CHAR(18) PRIMARY KEY NOT NULL,
Estimated_Delivery_Date DATE NOT NULL,
Shipping_Fee DECIMAL(4, 2),
delivery_method_code CHAR(2),
FOREIGN KEY (delivery_method_code) REFERENCES DeliveryMethod(Delivery_Method_Code)
);
```

```
124 •   SELECT * FROM `Lacoste Online Sales`.`TrackingInfo`;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Tracking_Number	Estimated_Delivery_Date	Shipping_Fee	delivery_method_code
0FF01I876292562045	2022-01-11	0.00	1
1RF07O673292562123	2022-01-22	0.00	2
1WP09Y871292562798	2022-01-05	20.00	2
1ZE98W210304252259	2022-01-12	0.00	1
2PZ09Y673292562798	2022-01-31	0.00	1
2UH01D03434223767	2022-01-08	0.00	1
3ER01B03434256012	2022-01-15	0.00	1
3GT01I876116761920	2022-01-22	0.00	1
3RF91J030979231210	2022-01-10	0.00	1
4SP01L03469654083	2022-01-24	0.00	1
5NS01I02411265408	2022-01-12	25.00	1
7RT34E05611276349	2022-01-07	0.00	1

TrackingInfo 6 | Apply

8. Create Table – "**Order Info**" with Order Code as Primary Key & Tracking No, Payment Method Code & Return Eligibility Code as Foreign Key.

```
CREATE TABLE OrderInfo
(
Order_Code CHAR(10) PRIMARY KEY NOT NULL,
Order_Date DATE NOT NULL,
tracking_number CHAR(18),
payment_method_code CHAR(5),
return_eligibility_code CHAR(2),
FOREIGN KEY (tracking_number) REFERENCES TrackingInfo(Tracking_Number),
FOREIGN KEY (payment_method_code) REFERENCES PaymentMethod(Payment_Method_Code),
FOREIGN KEY (return_eligibility_code) REFERENCES ReturnEligibility(Return_Eligibility_Code)
);
```

```
151 •   SELECT * FROM `Lacoste Online Sales`.`OrderInfo`;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Order_Code	Order_Date	tracking_number	payment_method_code	return_eligibility_code
4031249	2022-01-02	2UH01D03434223767	1	1
4153270	2022-01-03	7RT34E05611276349	1	1
4256123	2022-01-04	1WP09Y871292562798	2	1
4270732	2022-01-05	3RF91J030979231210	2	1
4471954	2022-01-07	0FF01I876292562045	1	1
4483288	2022-01-09	1ZE98W210304252259	1	1
4562256	2022-01-11	3ER01B03434256012	2	2
4585675	2022-01-12	5NS01I02411265408	1	1
4630985	2022-01-18	1RF07O673292562123	2	2
4645423	2022-01-18	3GT01I876116761920	1	1
4691974	2022-01-20	4SP01L03469654083	1	2
5013462	2022-01-22	9SD01I124145231901	1	1

OrderInfo 7 | Apply

9. Create Table – "**Product Order**" with PO_Id as Primary Key and, Order Code & Product Code as Foreign Key.

```
CREATE TABLE ProductOrder
(
```

```
PO_id INTEGER AUTO_INCREMENT PRIMARY KEY,
order_code CHAR(10),
product_code CHAR(5),
Quantity SMALLINT NOT NULL,
FOREIGN KEY (order_code) REFERENCES OrderInfo(Order_Code),
FOREIGN KEY (product_code) REFERENCES Product(Product_Code)
);
```

```
212 •  SELECT * FROM `Lacoste Online Sales`.`ProductOrder`;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

PO_id	order_code	product_code	Quantity
1	4483288	3HBM	1
2	4483288	0HBM	1
3	4256123	7HBM	1
4	5111342	29HBM	1
5	5111342	2HBM	1
6	4630985	1HBM	1
7	4471954	9HBM	1
8	4645423	4HBM	1
9	4645423	01HBM	1
10	4645423	22HBM	1
11	4645423	14HBM	1
12	5013462	25HBM	2

ProductOrder 8 × Apply

10. Create Table – "**Customer Order**" with CO_Id as Primary Key and, Order Code & Customer ID as Foreign Key.

```
CREATE TABLE CustomerOrder
(
CO_id INTEGER AUTO_INCREMENT PRIMARY KEY,
customer_id CHAR(5),
order_code CHAR(10),
FOREIGN KEY (customer_id) REFERENCES Customer(Customer_ID),
FOREIGN KEY (order_code) REFERENCES OrderInfo(Order_Code)
);
```

```
250 •  SELECT * FROM `Lacoste Online Sales`.`CustomerOrder`;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

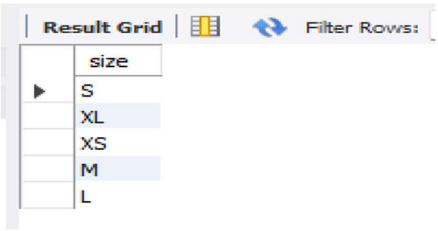
CO_id	customer_id	order_code
1	1	4483288
2	2	4256123
3	3	5111342
4	4	4630985
5	5	4471954
6	6	4645423
7	7	5013462
8	8	4585675
9	9	4691974
10	10	4562256
11	11	4031249
12	12	4270732

CustomerOrder 9 × Apply

SQL QUERIES TO ADDRESS BUSINESS QUESTIONS

1. Select distinct values of different sizes from the table.

```
SELECT distinct(size) from `Transaction log`;
```

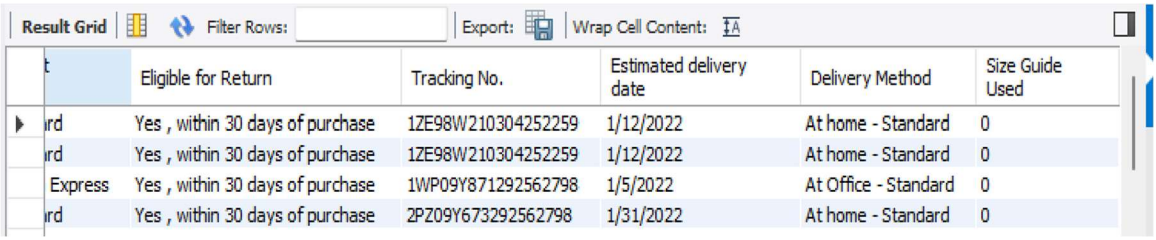


Result Grid | Filter Rows:

size
S
XL
XS
M
L

2. Alter table to add columns 'Size guide used' (values - Yes, No).

```
ALTER TABLE  `Transaction log`

ADD COLUMN `Size Guide Used` boolean default false NOT NULL;

SELECT * FROM `Transaction log`;
```

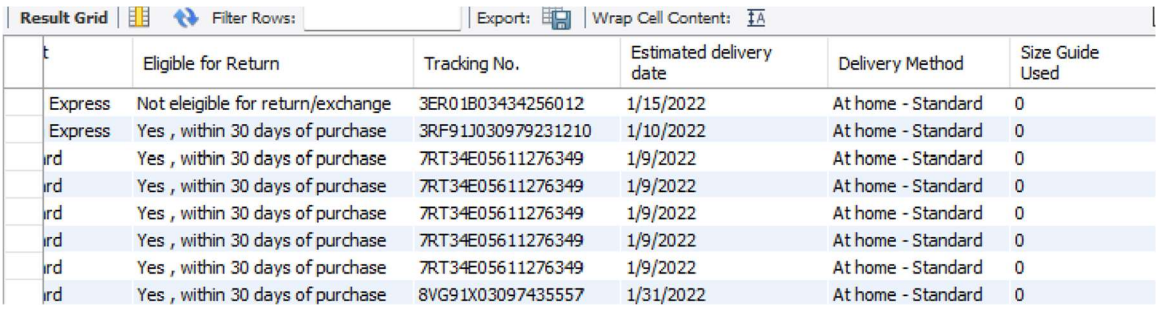


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

t	Eligible for Return	Tracking No.	Estimated delivery date	Delivery Method	Size Guide Used
rd	Yes , within 30 days of purchase	1ZE98W210304252259	1/12/2022	At home - Standard	0
rd	Yes , within 30 days of purchase	1ZE98W210304252259	1/12/2022	At home - Standard	0
Express	Yes , within 30 days of purchase	1WP09Y871292562798	1/5/2022	At Office - Standard	0
rd	Yes , within 30 days of purchase	2PZ09Y673292562798	1/31/2022	At home - Standard	0

3. Push the estimated delivery date for products ordered on 1/8/2022 by a day.

```
UPDATE  `Transaction log`

SET `Estimated delivery date` = '1/9/2022'

where `Estimated delivery date` = '1/8/2022';

SELECT * FROM `Transaction log`;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

t	Eligible for Return	Tracking No.	Estimated delivery date	Delivery Method	Size Guide Used
Express	Not eleigible for return/exchange	3ER01B03434256012	1/15/2022	At home - Standard	0
Express	Yes , within 30 days of purchase	3RF91J030979231210	1/10/2022	At home - Standard	0
rd	Yes , within 30 days of purchase	7RT34E05611276349	1/9/2022	At home - Standard	0
rd	Yes , within 30 days of purchase	7RT34E05611276349	1/9/2022	At home - Standard	0
rd	Yes , within 30 days of purchase	7RT34E05611276349	1/9/2022	At home - Standard	0
rd	Yes , within 30 days of purchase	7RT34E05611276349	1/9/2022	At home - Standard	0
rd	Yes , within 30 days of purchase	7RT34E05611276349	1/9/2022	At home - Standard	0
rd	Yes , within 30 days of purchase	8VG91X03097435557	1/31/2022	At home - Standard	0

4. What is the volume of sales for Product Color - White?

```
SELECT SUM(`Quantity`) from `Transaction log` where
`Color` = 'White';
```

Result Grid | Filter Rows:

	SUM(`Quantity`)
▶	7

5. Name the top three customers who bought the highest number of products?

```
SELECT `Customer Name`, SUM(`Quantity`) from `Transaction log` group by `Customer Name`
order by SUM(`Quantity`) DESC LIMIT 3;
```

Result Grid | Filter Rows:

	Customer Name	SUM(`Quantity`)
▶	Rudi Josh	5
	Ken Neil	4
	Karen Teal	3

6. Print product description of all the products which are in blue color

```
SELECT*FROM Product WHERE `Color` = "Blue";
```

Result Grid | Filter Rows: | Edit: | Export/Import:

	Product_Code	Product_Description	Product_Cost	Size	Color
▶	18HBM	Mens Stretch Cotton Chino Pants	89.50	L	Blue
	22HBM	Mens Striped Fleece Jogging Pants	76.99	XL	Blue
	3HBM	Mens Badge Zip Pocket Hooded Sweatshirt	81.99	S	Blue
	41HBM	Womens Cotton Turtleneck	48.99	XS	Blue
*	NULL	NULL	NULL	NULL	NULL

7. Print products that are unisex.

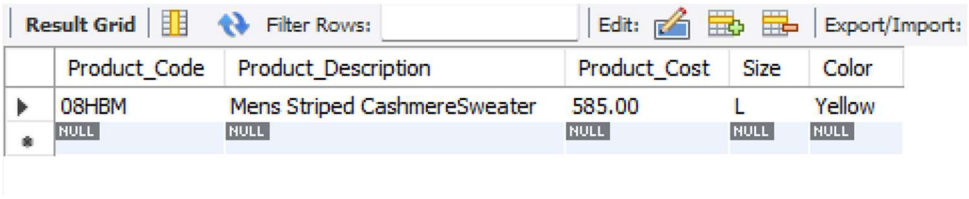
```
SELECT*FROM Product WHERE `Product_Description` LIKE 'Unisex%';
```

Result Grid | Filter Rows: | Edit: | Export/Import:

	Product_Code	Product_Description	Product_Cost	Size	Color
▶	00BBM	Unisex L. 12. 12 Adjustable Face Mask	45.00	L	Pink
	78HBM	Unisex L. 12. 12 Face Masks 3-Pack	34.99	S	Black
*	NULL	NULL	NULL	NULL	NULL

8. List the most expensive product in store

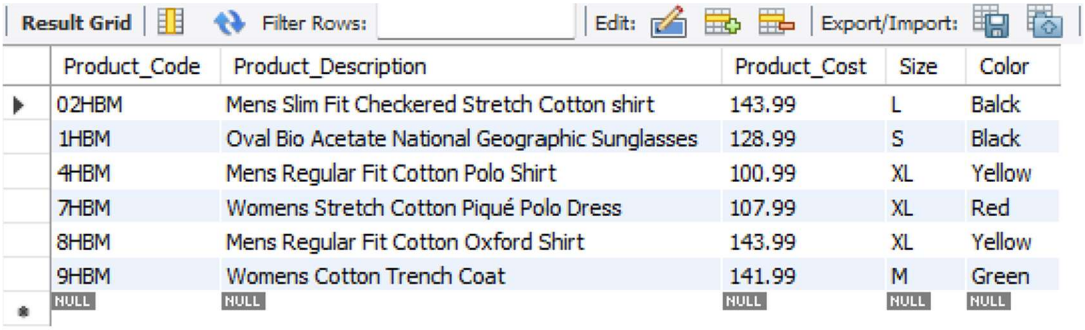
```
SELECT MAX(Product_Cost) FROM Product;

SELECT*FROM Product Order By Product_cost DESC LIMIT 1;
```



Product_Code	Product_Description	Product_Cost	Size	Color
08HBM	Mens Striped CashmereSweater	585.00	L	Yellow
NULL	NULL	NULL	NULL	NULL

9. List products that cost from 100 – 200

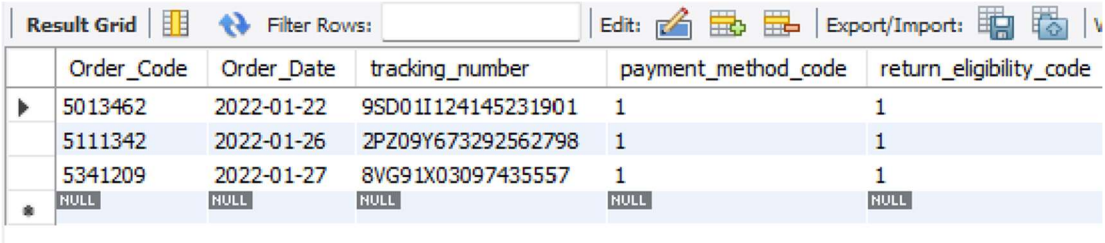
```
SELECT*FROM Product WHERE `Product_cost` BETWEEN 100 AND 200;
```



Product_Code	Product_Description	Product_Cost	Size	Color
02HBM	Mens Slim Fit Checkered Stretch Cotton shirt	143.99	L	Balck
1HBM	Oval Bio Acetate National Geographic Sunglasses	128.99	S	Black
4HBM	Mens Regular Fit Cotton Polo Shirt	100.99	XL	Yellow
7HBM	Womens Stretch Cotton Piqué Polo Dress	107.99	XL	Red
8HBM	Mens Regular Fit Cotton Oxford Shirt	143.99	XL	Yellow
9HBM	Womens Cotton Trench Coat	141.99	M	Green
NULL	NULL	NULL	NULL	NULL

10. Print product code of all the products that were ordered after 20 Jan 2022.

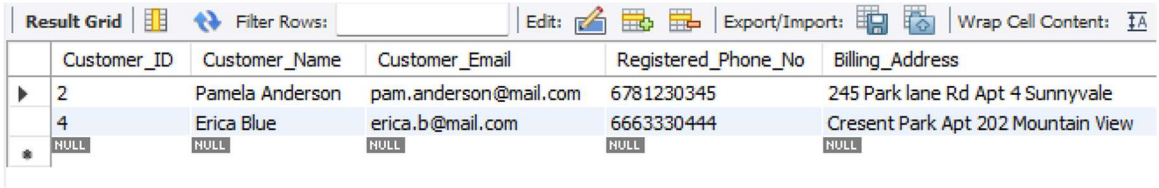
```
SELECT*FROM Orderinfo WHERE `Order_date`> '2022-01-20';
```



Order_Code	Order_Date	tracking_number	payment_method_code	return_eligibility_code
5013462	2022-01-22	9SD01I124145231901	1	1
5111342	2022-01-26	2PZ09Y673292562798	1	1
5341209	2022-01-27	8VG91X03097435557	1	1
NULL	NULL	NULL	NULL	NULL

11. List details of customers who have different billing and shipping address

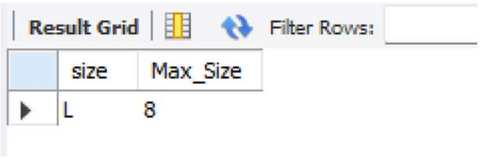
```
SELECT * FROM Customer WHERE Billing_Address != Shipping_address;
```



Customer_ID	Customer_Name	Customer_Email	Registered_Phone_No	Billing_Address
2	Pamela Anderson	pam.anderson@mail.com	6781230345	245 Park lane Rd Apt 4 Sunnyvale
4	Erica Blue	erica.b@mail.com	6663330444	Cresent Park Apt 202 Mountain View
NULL	NULL	NULL	NULL	NULL

12. What size of product was sold most

```
SELECT size, count(SIZE) AS Max_Size
FROM product GROUP BY Size
ORDER BY Max_Size DESC LIMIT 1;
```

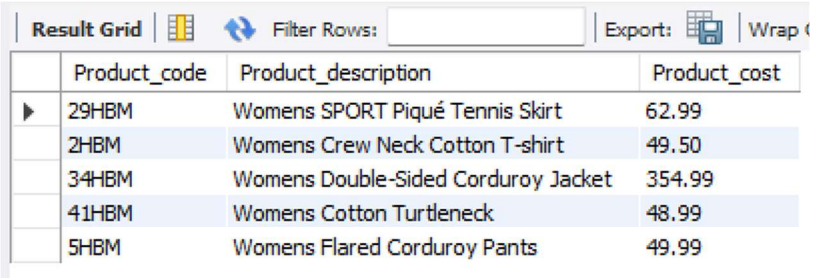


Result Grid | Filter Rows:

size	Max_Size
L	8

13. Print product cost of all the XS sized products

```
SELECT Product_code, Product_description, Product_cost from Product WHERE Size = 'XS';
```

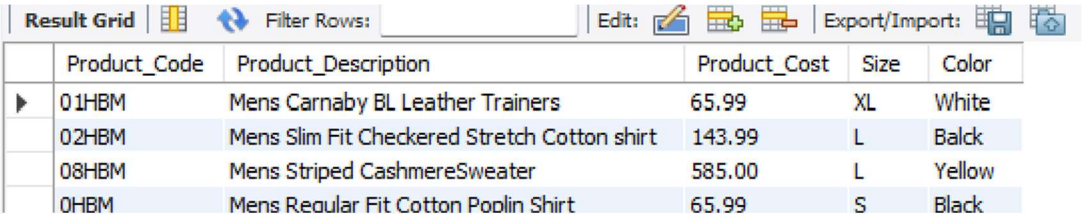


Result Grid | Filter Rows: | Export: | Wrap

Product_code	Product_description	Product_cost
29HBM	Womens SPORT Piqué Tennis Skirt	62.99
2HBM	Womens Crew Neck Cotton T-shirt	49.50
34HBM	Womens Double-Sided Corduroy Jacket	354.99
41HBM	Womens Cotton Turtleneck	48.99
5HBM	Womens Flared Corduroy Pants	49.99

14. Print all men's merchendise sold

```
SELECT * FROM product WHERE Product_Description like ('Mens%');
```



Result Grid | Filter Rows: | Edit: | Export/Import:

Product_Code	Product_Description	Product_Cost	Size	Color
01HBM	Mens Carnaby BL Leather Trainers	65.99	XL	White
02HBM	Mens Slim Fit Checkered Stretch Cotton shirt	143.99	L	Balck
08HBM	Mens Striped CashmereSweater	585.00	L	Yellow
0HBM	Mens Regular Fit Cotton Poplin Shirt	65.99	S	Black

WITH JOINS

Joining tables - Order Info & Product Order

```
SELECT o.Order_Code, Order_Date, Tracking_Number, Payment_Method_code, Return_eligibility_code,
PO_id,

product_code, quantity FROM OrderInfo O

LEFT JOIN ProductOrder p

ON O.Order_Code = P.Order_Code;
```

Order_Code	Order_Date	Tracking_Number	Payment_Method_code	Return_eligibility_code	PO_id	product_code	quantity
4031249	2022-01-02	2UH01D03434223767	1	1	16	34HBM	1
4031249	2022-01-02	2UH01D03434223767	1	1	17	41HBM	1
4031249	2022-01-02	2UH01D03434223767	1	1	18	5HBM	1
4031249	2022-01-02	2UH01D03434223767	1	1	41	34HBM	1
4031249	2022-01-02	2UH01D03434223767	1	1	42	41HBM	1

Joining tables - **Customer – customer order – order info - Tracking Info** to generate list of customers & their details who opted for paid shipping

```
SELECT c.Customer_Name, c.Shipping_Address, ti.Shipping_Fee

FROM customer c

LEFT JOIN customerorder co ON c.customer_id = co.customer_id

LEFT JOIN orderinfo oi ON co.order_code = oi.order_code

LEFT JOIN trackinginfo ti ON oi.tracking_number = ti.tracking_number

WHERE ti.Shipping_Fee > 0;
```

Customer_Name	Shipping_Address	Shipping_Fee
Pamela Anderson	Mindspace Omega-Building C San Jose	20.00
Pamela Anderson	Mindspace Omega-Building C San Jose	20.00
Susan Hay	Spring Creek Apt 121 Stevens Creek	25.00
Susan Hay	Spring Creek Apt 121 Stevens Creek	25.00

Joining tables - **customer - customerorder – orderinfo – payment method** to generate list of customers who have made payment through Credit card

```
SELECT c.Customer_Name, c.Registered_Phone_No, c.Billing_Address, pt.payment_method_description

FROM customer c

LEFT JOIN customerorder co ON c.customer_id = co.customer_id

LEFT JOIN orderinfo oi ON co.order_code = oi.order_code

LEFT JOIN paymentmethod pt ON oi.payment_method_code= pt.payment_method_code

WHERE pt.payment_method_code =1;
```

Customer_Name	Registered_Phone_No	Billing_Address	payment_method_description
Kylie John	6664440555	Palm Court House No 456 Los Gatos	Credit Card
Kylie John	6664440555	Palm Court House No 456 Los Gatos	Credit Card
Rudi Josh	6780901004	1040 Valleyfair Road Milpitas	Credit Card
Rudi Josh	6780901004	1040 Valleyfair Road Milpitas	Credit Card
Ashley Turner	6665550666	245 Park lane Rd Apt 21 Palo Alto	Credit Card
Ashley Turner	6665550666	245 Park lane Rd Apt 21 Palo Alto	Credit Card

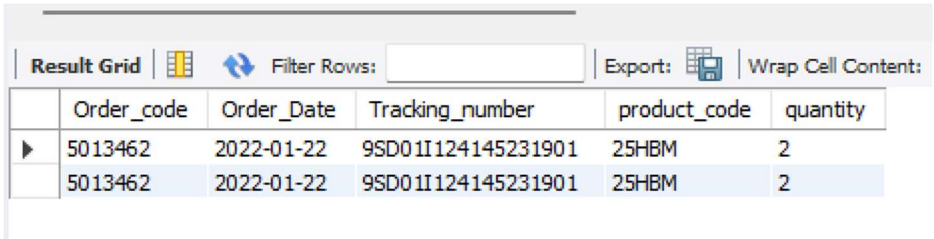
Joining tables - **customer – customerorder – product order – product** to generate list of customers who have purchased blue color merchandise?

```
SELECT c.Customer_Name, c.Registered_Phone_No, c.Shipping_Address, pr.product_description,
pr.product_cost, pr.color

FROM customer c

LEFT JOIN customerorder co ON c.customer_id = co.customer_id

LEFT JOIN productorder po ON co.order_code = po.order_code

LEFT JOIN product pr ON po.product_code = pr.Product_Code

WHERE pr.color = 'blue';
```

Customer_Name	Registered_Phone_No	Shipping_Address	product_description	product_c
Rudi Josh	6780901004	1040 Valleyfair Road Milpitas	Mens Stretch Cotton Chino Pants	89.50
Rudi Josh	6780901004	1040 Valleyfair Road Milpitas	Mens Stretch Cotton Chino Pants	89.50
Rudi Josh	6780901004	1040 Valleyfair Road Milpitas	Mens Stretch Cotton Chino Pants	89.50
Rudi Josh	6780901004	1040 Valleyfair Road Milpitas	Mens Stretch Cotton Chino Pants	89.50
Ken Neil	6666660777	Northseal Sq H No 220 Cupertino	Mens Striped Fleece Jogging Pants	76.99
Ken Neil	6666660777	Northseal Sq H No 220 Cupertino	Mens Striped Fleece Jogging Pants	76.99

Joining tables – **Order Info - Product Order** to print tracking number & product code of items which are ordered on 22 Jan 2022.

```
SELECT OI.Order_code, oi.Order_Date, oi.Tracking_number, p.product_code, p.quantity

FROM OrderInfo oi

LEFT JOIN ProductOrder p ON oi.ORDER_CODE = p.order_code

WHERE oi.ORDER_DATE = '2022-01-22';
```



Order_code	Order_Date	Tracking_number	product_code	quantity
5013462	2022-01-22	9SD01I124145231901	25HBM	2
5013462	2022-01-22	9SD01I124145231901	25HBM	2

VIEWS AND SUBQUERIES

Creating views using INNER JOIN

```
CREATE VIEW Tracking AS

SELECT T.Tracking_Number, Estimated_Delivery_Date, Shipping_Fee, Order_code, Order_date FROM
TrackingInfo T

INNER JOIN OrderInfo O

ON T.Tracking_Number = O.Tracking_Number;

## Display all details present in the view

SELECT * FROM Tracking;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Tracking_Number	Estimated_Delivery_Date	Shipping_Fee	Order_code	Order_date
0FF01I876292562045	2022-01-11	0.00	4471954	2022-01-07
1RF07O673292562123	2022-01-22	0.00	4630985	2022-01-18
1WP09Y871292562798	2022-01-05	20.00	4256123	2022-01-04
1ZE98W210304252259	2022-01-12	0.00	4483288	2022-01-09
2PZ09Y673292562798	2022-01-31	0.00	5111342	2022-01-26

Creating views using LEFT JOIN

```
CREATE VIEW Order_And_Product AS

SELECT o.Order_Code, Order_Date, Tracking_Number, Payment_Method_code, Return_eligibility_code,

product_code, quantity FROM OrderInfo O

LEFT JOIN ProductOrder p

ON O.Order_Code = P.Order_Code;

## Display all details present in the view

SELECT * FROM Order_And_Product;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Order_Code	Order_Date	Tracking_Number	Payment_Method_code	Return_eligibility_code	product_code	quantity
4031249	2022-01-02	2UH01D03434223767	1	1	34HBM	1
4031249	2022-01-02	2UH01D03434223767	1	1	41HBM	1
4031249	2022-01-02	2UH01D03434223767	1	1	5HBM	1
4031249	2022-01-02	2UH01D03434223767	1	1	34HBM	1
4031249	2022-01-02	2UH01D03434223767	1	1	41HBM	1
4031249	2022-01-02	2UH01D03434223767	1	1	5HBM	1
4153270	2022-01-03	7RT34E05611276349	1	1	50HBM	1

SQL - SubQuery

I have joined different normalized tables using sub-query to answer business questions and to understand how sub query works and what role operators like 'IN' play in sub query. From the joins that we have performed above we can see how similar output, that is derived from joins can be derived from sub-queries.

Using sub-query to list all the products that are not eligible for return (Queried through tables: Product – product order – order info – return eligibility).

```
SELECT product_description, product_cost, size, color from product WHERE Product_Code IN(
        SELECT Product_code FROM productorder WHERE Order_code IN(
                (SELECT order_code FROM orderinfo WHERE return_eligibility_code = 2)
        )
);
```

	product_description	product_cost	size	color
▶	Unisex L. 12. 12 Face Masks 3-Pack	34.99	S	Black
	Oval Bio Acetate National Geographic Sunglasses	128.99	S	Black
	Unisex L. 12. 12 Adjustable Face Mask	45.00	L	Pink

Using sub-query to answer how many customers selected "at-office standard" delivery method (Queried through tables: Customer- customerorder – orderinfo- tracking info)

```
SELECT customer_name,customer_email, shipping_address from customer WHERE Customer_ID IN(
        SELECT Customer_ID FROM customerorder WHERE order_code IN(
                SELECT order_code FROM orderinfo WHERE tracking_number IN(
                        (SELECT tracking_number FROM trackinginfo WHERE delivery_method_code = 2)
                )
        )
);
```

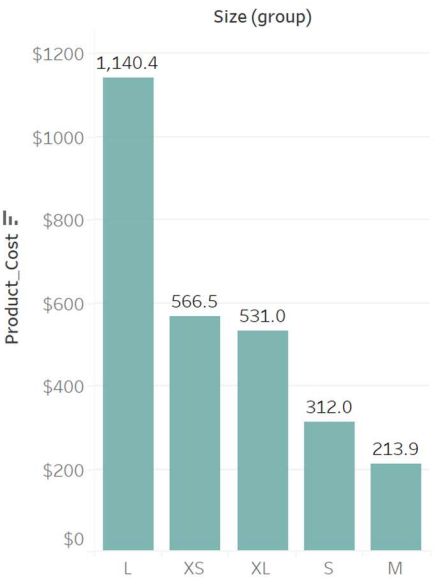
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	customer_name	customer_email	shipping_address
▶	Pamela Anderson	pam.anderson@mail.com	Mindspace Omega-Building C San Jose
	Erica Blue	erica.b@mail.com	North San Jose Tech Park Building21

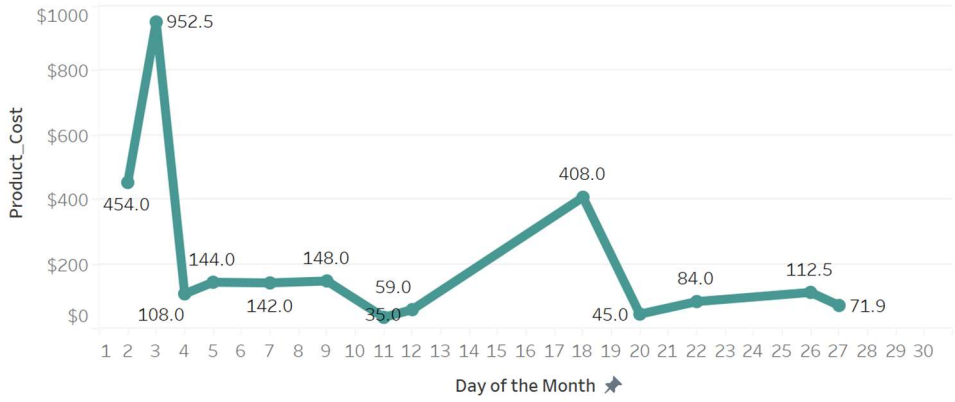
Tableau Report

City Wise Sales

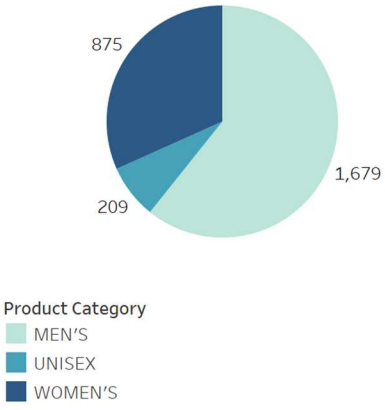
Total Sale Per Size



Daily Sales Trend



Diversity In Products Sold



CONCLUSION

The online sales management system is developed to provide additional insights on sales through online platform and analyzes sales patterns to maximize profit. The model, if implied on larger base could result in providing useful stats for Lacoste which will help them strategize their marketing, advertising, target customer & production to meet demand. A centralized database system can help Lacoste to improve their efficiency in all aspects.

As a part of this project, I learnt the foundational concepts of data modelling and relational database management systems (R-DBMS) and could apply different features and aspects of it to optimally represent, store and manage the data involved in the business scenario. I also learnt the use of SQL language to store, retrieve, manipulate, and analyze the data from the databases. Through the fundamental constructs of SQL, I was able to answer the proposed business questions from Lacoste perspective thoroughly.

With advanced features of SQL, this project can be made more sophisticated, by using triggers and stored procedure to make it user friendly. Also, capturing more attributes in the stored information such as customer preferences, data across different states, multiple payment methods, etc. could make the analysis more interesting and provide further insightful results.