Target Enterprises
A Perfect Store~Expect More. Pay Less

TARGET ENTERPRISE PVT. LTD.

Team 4

By

Abhishek Rai

Vidhi Shah

Shashwat Khambhatta

TABLE OF CONTENTS

INTRODUCTION

Target Enterprise Pvt. Ltd., a mail order company exporting variety of luxury, consumer and novelty items. This company is small scale growing company as it has 27 employees at start. These 27 employees are divided among different departments. There are five departments namely marketing, accounting, order processing, dispatch, and purchasing. Each department has various employees and is managed by a Manager. Their main working depends upon the orders received from the customers. Customers can order items by mail, phone or fax. There are different ways by which the order is placed, shipped and delivered. In this process of receiving the orders and delivering it to the customer various departments i.e order process, purchasing, accounts, marketing and dispatch were included at various stages.

BUSINESS PROBLEM

Business Problem:

There are numerous problems which are faced by the company and main reason leading to these problems is the outdated and traditional way the company works. The problems arising due to these are:

1. Most of the processes like order process, dispatch, etc. is done manually: Whether it is an order from post or fax, the receiving clerk have to check the details manually. Once finishing checking the order, he would initial the top right corner and stamp the order form with a date stamp. Thee telephonic orders are also taken manually by the clerks. The payment executed by the customer is recorded by the hand too. Orders received by fax are ended up in queries tray. There is a clerk whole sole responsibility is to follow up on queries. Each clerk keeps two lists, first is of checks and postal orders and other is other list is having the card details of the customers. The problem with this approach is that the system is time consuming and takes lot of work to execute it. Say if a clerk misplaces or makes a different order, the whole process of that order has to be started from the start and thus causing more expense on the way.

2. All the records are paper based: All the forms carrying complex information are in paper form and are to be filled in hand. For example, clerks who process the payment details need to enter the customer details on a list. In order to know if the order is completed properly, the clerk has to go through the forms manually. Plus, paper based documents occupy much storage space as more and more orders come in. It takes up a lot of time to retrieve after a long time because of lot of storage taken to store.

3. Inadequate software to provide effective and operative accounting reports: The company is not able to segment the data for different customers due to not keeping their customer

database in line. Due to not using the system for their records, the change in phone numbers and address of the customers cannot be monitored resulting into delivering duplicate promotional catalogs.

4. Outdated database system and other technologies: dBase 3 system: The company lacked an integrated and up-to-date information system to help people managing the accounting, finance, management and purchasing activities. Thus it is very hard for each department to control cost, corporately operate routing task and making right business decision.

SCOPE AND FOCUS OF THE DATABASE SOLUTION

In order to survive the various business problems, new measures have to be taken to find ways where the company can manage the customer database and operate it smoothly. The database could be designed in such a way that all the customers and order information can be stored and tracked. The validation can be applied to new incoming orders and new payments processed. Hence the scope of the database should focus on customers, orders and payments of new incoming orders and also the existing order information. Reports must be fetched at regular intervals to educate the company about the status of items, order and customer processes. Hence a flexible database can be built to reduce the pressure on the employees by shifting the processing using a database rather than manually doing the same process.

Requirements of the business:

1. Online ordering and dispatching system.
2. An inventory management System for better marketing, inventory and account reports.
3. Auto updates on inventory and order status in every 24 hrs and provide day to day report.
4. Update the current database to centralized database like Microsoft SQL server 2014.
   Store customer information and link it with their purchase history.

PARENTHESIZED METHODS

Below mentioned are the tables present in the Target Enterprise Ltd. database with all the attributes described.

Below mentioned are the tables present in the Target Enterprise Ltd. database with all the attributes described.

1. **ORDER_FORM** (Order_ID, *Customer_ID*, *Item_ID*, *Clerk_ID*, Date_Ordered, Delivery_Type, Order_Method, Payment_Type, Total_Price, Status_Code, Item_Status)
   This table contains details of all the orders placed by various customers and the various items in each order. It contains all the required parameters associated to that order like delivery type, payment method, price, status of the order, etc.
   Order_ID is the primary key and Customer_ID, Item_ID, Clerk_ID & Status_code represent the foreign key.

2. **CUSTOMER** (Customer_ID, Firstname, Lastname, Phone_Number, Email, Fax, Street, City, State, Zip)
   The CUSTOMER table contains the records of all the customers who place an order to the clerks.
   It contains details like their names and addresses. Customer_ID is the primary key in this table.

3. **INVENTORY** (Item_ID, *Store_ID*, Category, Sub-Category, Price, Color, Size, Item_Status, Item_Qty_Available, Item_Qty_Sold)
   The Inventory table holds the details of all the items available in the inventory along with their description like color, price, size, etc. It also contains how much each item is available and the quantity sold.
   Item_ID is the primary key and Store_ID acts as foreign key.

4. **STORE** (Store_ID, *Item_ID*, Phone, Fax, Email, Street, City, State, Zip)
   This table contains the store details like the various Target stores located at different locations. Their address and contact details.
   Store_ID is the primary and Item_ID is the foreign.

5. **SHIPPING** (Shipping_ID, *Order_ID*, Ship_Street, Ship_City, Ship_State, Ship_Zip, Ship_Date)
   The SHIPPING table contains the details of where the items are to be shipped for example the shipping address and the shipping date.
   Shipping_ID is the primary key and Order_Id stands as the foreign key.

6. **CLERK** (Clerk_ID, *Dept_ID, Store_ID*, Clerk_Type, Clerk_Name, Clerk_Dept)
   This table holds the details of all the clerks present in the sytem. Their name,type and department.
   Clerk_ID is the primary key. Dept_ID, Store_ID are the respective foreign keys.

7. **ORDER_STATUS** (Status_Code, Status_Desc)

This table contains the details of all the statuses of the order like new, completed, incomplete, pending and cancelled. This is a static look-up table.
Status_Code is the primary key here.

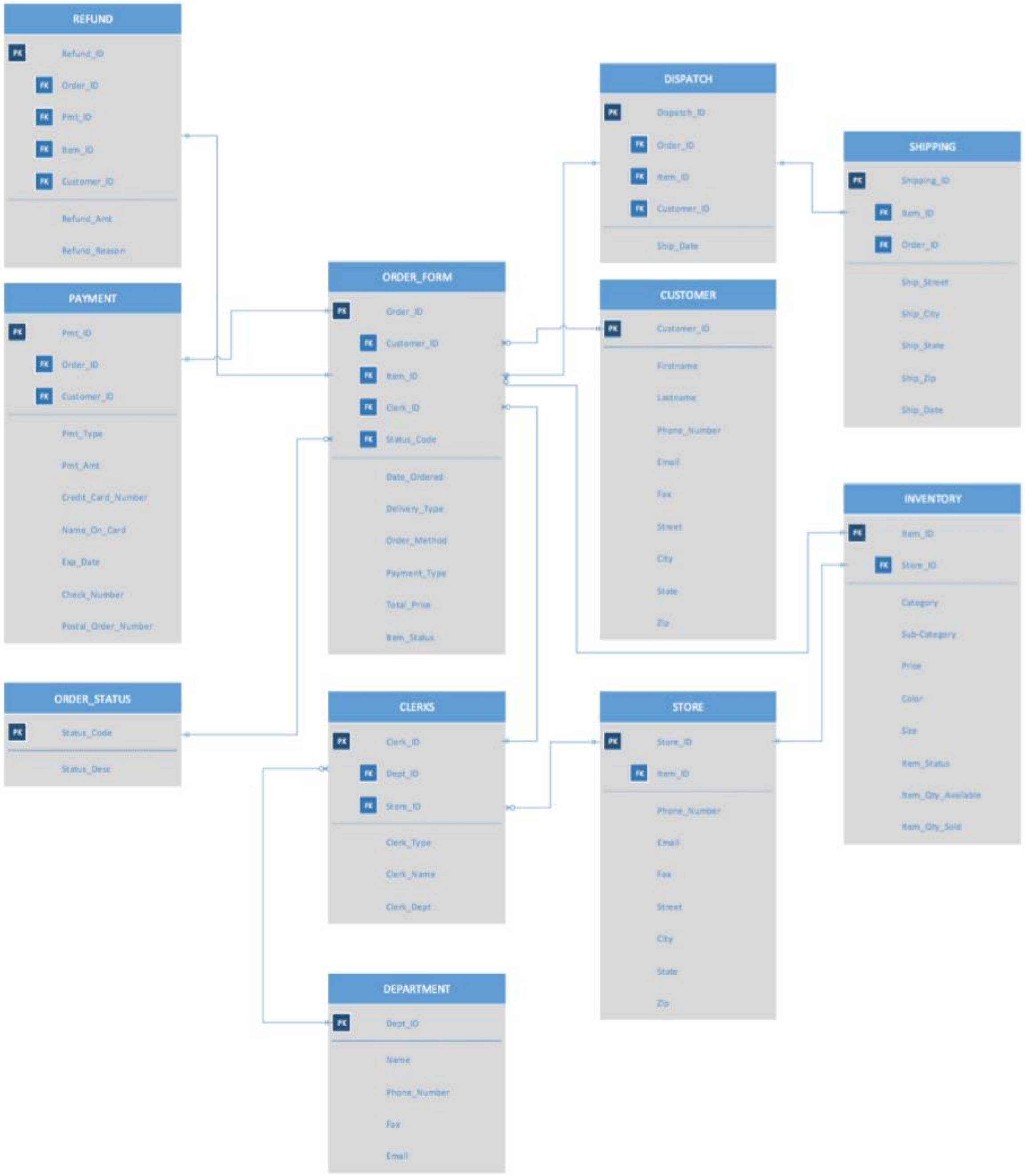
8. **DEPARTMENT** (Dept_ID, Dept_Name, Dept_Phone, Dept_Fax, Dept_Email)
   The DEPARTMENT table holds the department details of all the five departments as stated in the requirement. It contains their names and contact details. Here the primary key is Dept_ID.

9. **DISPATCH** (Dispatch_ID, *Order_ID, Item_ID, Customer_ID*, Ship_Date)
   The DISPATCH table contains the information about dispatch of an order.
   Dispatch_ID is the primary key whereas Order_ID, Item_ID and Customer_ID are the foreign keys.

10. **PAYMENT** (Pmt_ID, *Order_ID*, *Customer_ID*, Pmt_Type, Pmt_Amt, Credit_Card_Number, Name_On_Card, Exp_Date, Check_Number, Postal_Order_Number)
    This table contains the payment details of all the orders placed by the various customers.
    Pmt_ID is the primary key and Order_ID & Customer_ID are the foreign keys.

11. **REFUND** (Refund_ID, *Order_ID*, *Pmt_ID*, *Item_ID*, *Customer_ID,* Refund_Amt, Refund_Reason)
    This table contains all the details of all the orders which have been cancelled and a refund is to be made for them.
    Primary key: Refund_ID
    Foreign key: Order_ID, Pmt_ID, Item_ID, Customer_ID

NEW ER DIAGRAM



DESCRIPTION OF CHANGES

Changes:

1. Item_ID in the Inventory table and Product_ID in Product table are the same and they should be cut down to a one attribute itself.
2. Shipping date missing in Shipping table.
3. There must be an association table between Order_Form and Product table.
4. The Delivery address in Order_Form and shipping address in Shipping table are redundant attributes. Hence, must be cut down to a single attribute.
5. Payments must be tied to Shipping table.
6. Attributes must have shorter names as the names provided this time are a bit long ex- Customer_Address_Street must be simply Street.
7. Re-design the table by removing some of the redundant tables as well as redundant attributes.
8. Add association tables between tables having many to many cardinalities.

Updated System:

In the updated database design, we cut down some of the unnecessary attributes from the Order_Form table like the payment details were removed and were instead put into a separate Payment table which stored all the payment details for each and every order whatever may be the payment method. The Customer table was kept as it is except the names of the attributes were shortened. The Product table was removed from the database and its attributes were merged into the Inventory table thereby removing the redundancy between Item_ID and Product_ID. The Inventory table would also contain the amount of items sold and left. A new attribute with the name Item_Status was added to this table which would indicate the status of each item in the inventory for ex. Dispatched, returned, etc. The Shipping table was kept as it is. Instead of the Employee table we inserted a new table by the name Clerk which would be specific to all the clerks of the various departments. The Status table was renamed as Order_Status and the Department table was kept as it is. Apart from this two new tables were added namely Dispatch and Refund. The Dispatch table would be an intermediate table between the Order_Form and Shipping table as the items in the order which are ready for delivery would be recorded in the Dispatch table before they are shipped. The Refund table was added to take care of the items which needed to be returned or the cancelled orders for which the payment should be refunded to the customer.

LIST OF QUERIES

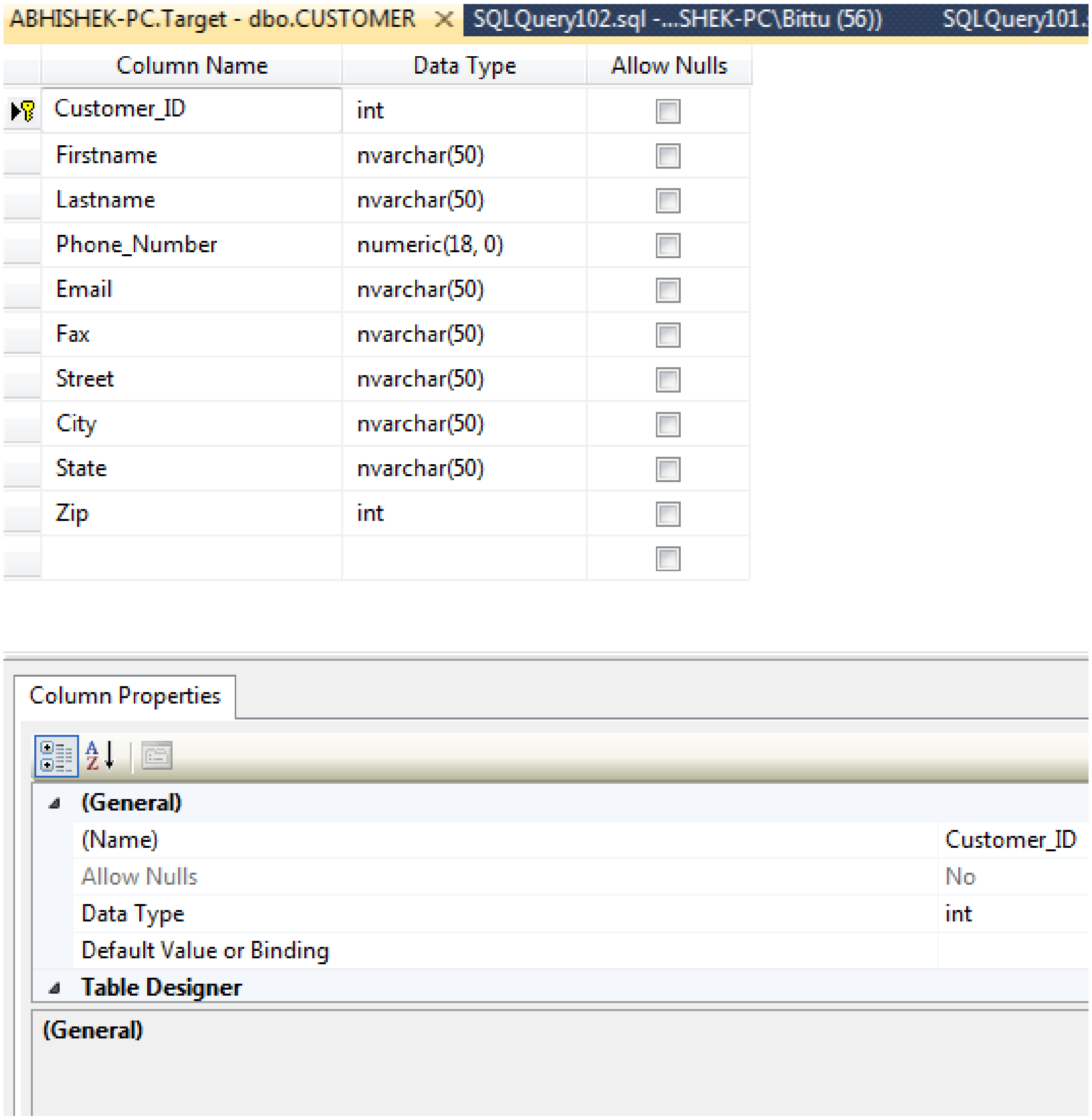
As mentioned in the INTRODUCTION section of this document, this section includes the SQL scripts developed in Project Part II during Target database design. The scripts have been included and submitted as a **separate text file** along with this project part III documentation. It contains details to create tables and insert values along with the necessary scripts, triggers and stored procedures required to run SQL commands.

NOTE: Since the SQL scripts have been designed on the basis of tables created from the entities described in Project Part II, kindly refer to Appendices Section for details of the E-R Model, design assumptions adopted while developing the E-R Model and also, documentation on different scenarios run successfully on the basis of database developed.

Sample Create Table and SQL Output:

```
CREATE TABLE CUSTOMER (

        Customer_ID    int          NOT NULL, IDENTITY(1,1),

        Firstname     nvarchar(50)    NOT NULL,

        Lastname      nvarchar(50)    NOT NULL,

        Phone_Number   numeric(18, 0)  NOT NULL,

        Email        nvarchar(50)    NOT NULL,

        Fax          nvarchar(50)    NOT NULL,

        Street       nvarchar(50)    NOT NULL,

        City         nvarchar(50)    NOT NULL,

        State        nvarchar(50)    NOT NULL,

        Zip          int          NOT NULL,

      Constraint    CustomerPK     PRIMARY KEY (Customer_ID)

      );
```

Output:


ABHISHEK-PC.Target - dbo.CUSTOMER
SQLQuery102.sql -...SHEK-PC\Bittu (56))
SQLQuery101.

Column Name	Data Type	Allow Nulls
Customer_ID	int	□
Firstname	nvarchar(50)	□
Lastname	nvarchar(50)	□
Phone_Number	numeric(18, 0)	□
Email	nvarchar(50)	□
Fax	nvarchar(50)	□
Street	nvarchar(50)	□
City	nvarchar(50)	□
State	nvarchar(50)	□
Zip	int	□
		□

Column Properties

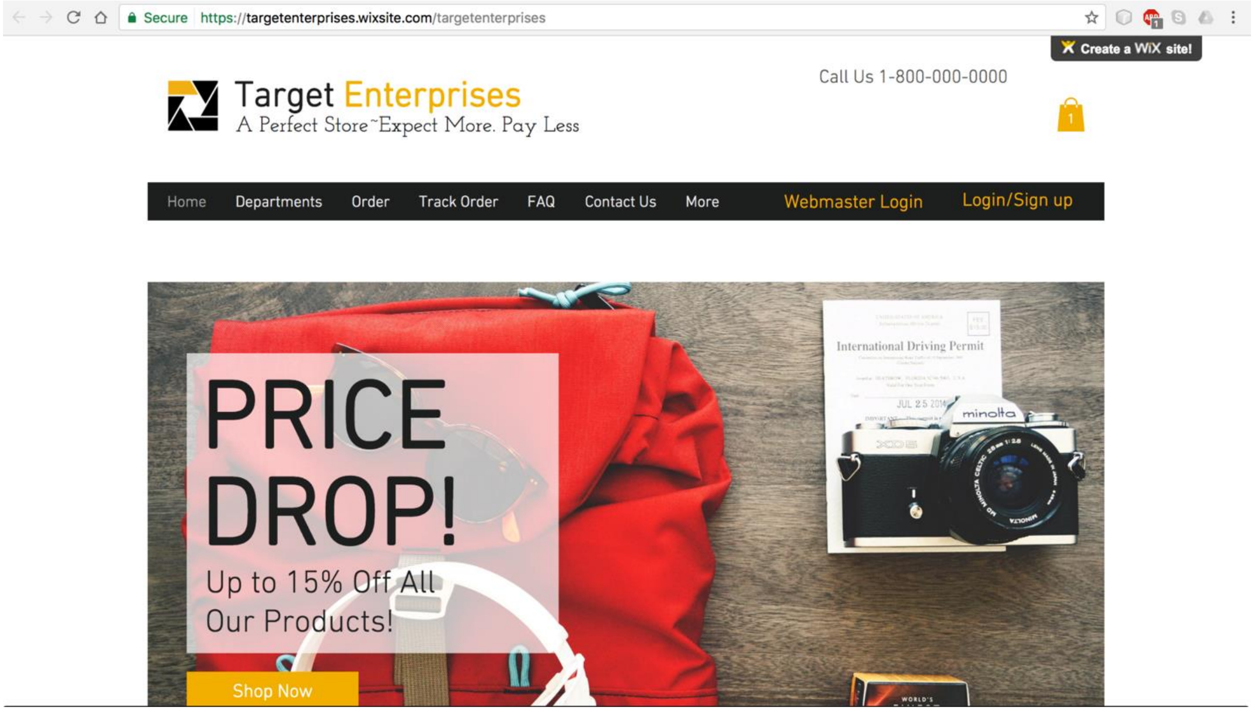
(General)

(Name)	Customer_ID
Allow Nulls	No
Data Type	int
Default Value or Binding	

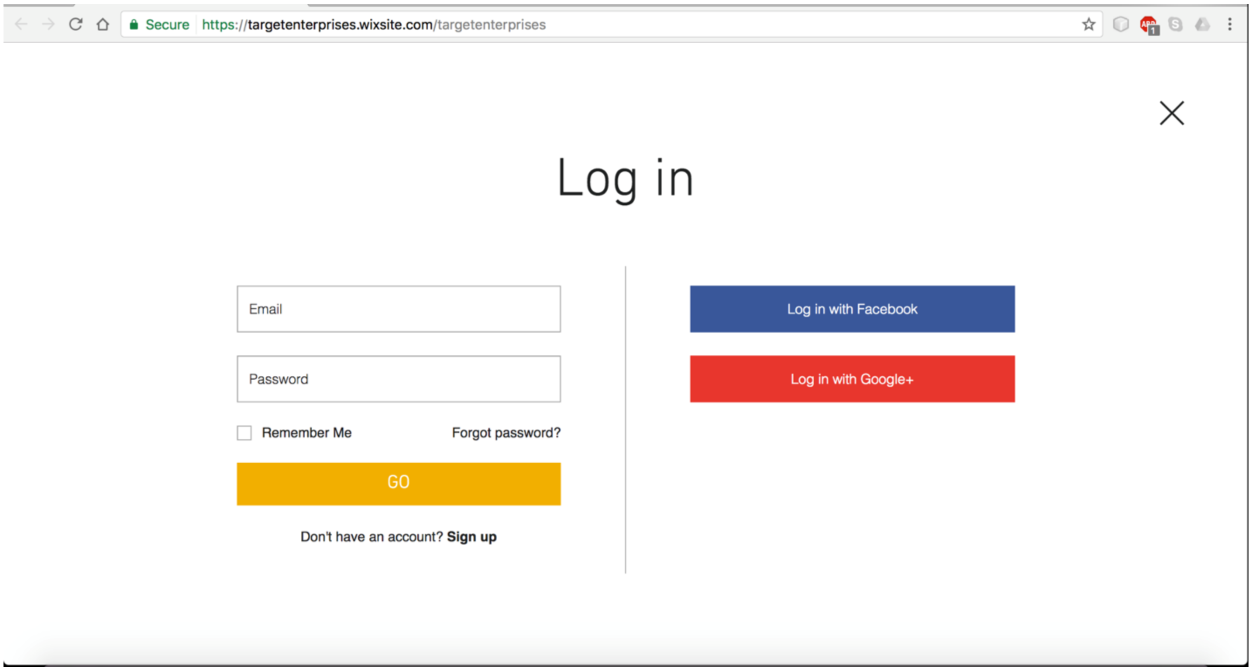
Table Designer

(General)

WEB PROTOTYPES SCREENSHOTS

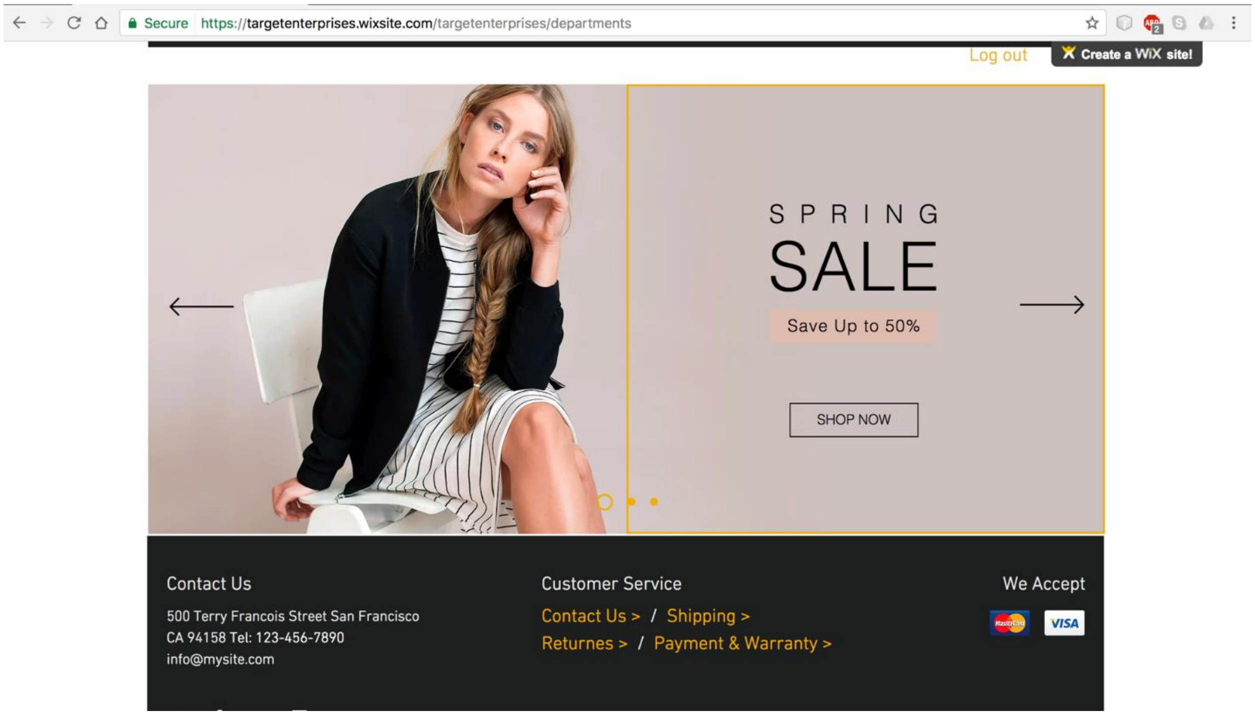


This is the main home page of the website of our company “Target Enterprises”. New offers that are trending would be advertised on this page.


Secure https://targetenterprises.wixsite.com/targetenterprises
Log in
Email
Password
□ Remember Me
Forgot password?
GO
Don't have an account? Sign up
Log in with Facebook
Log in with Google+

The clerk can login to the portal of our website by using various methods like login with

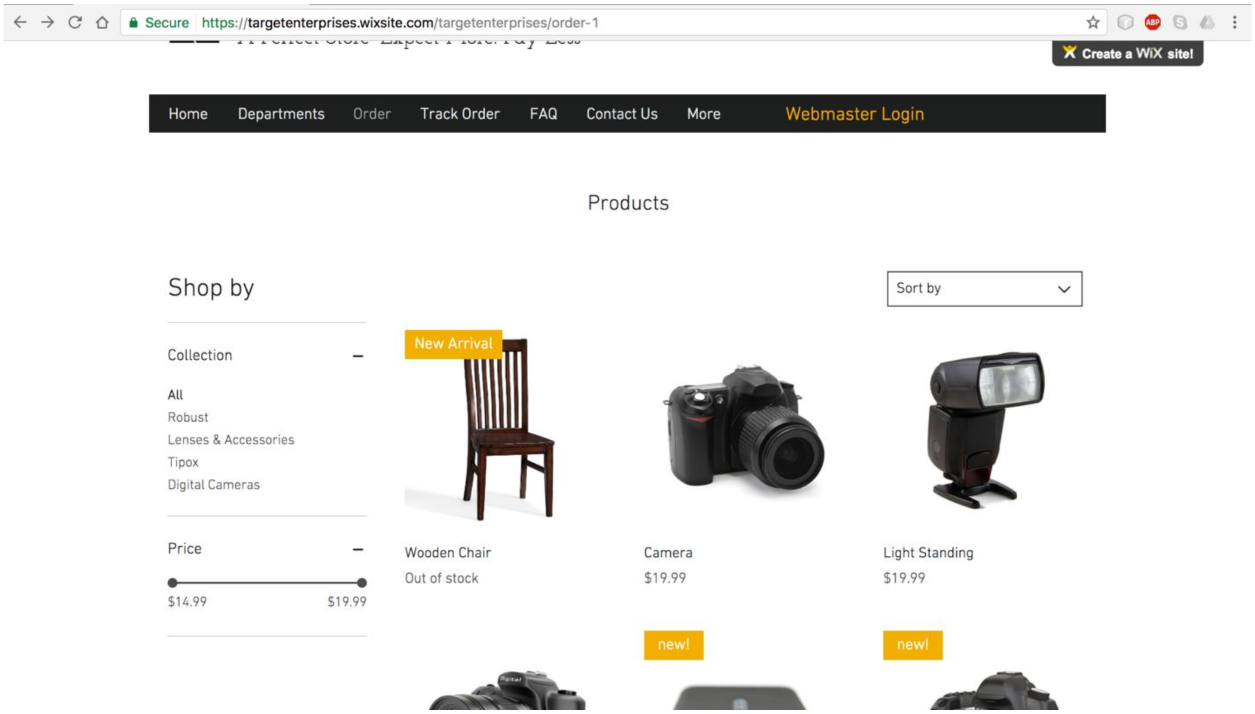
Facebook account, Google+ account or his own account on “Target Enterprises”. This login helps maintain data secure for the clerk.



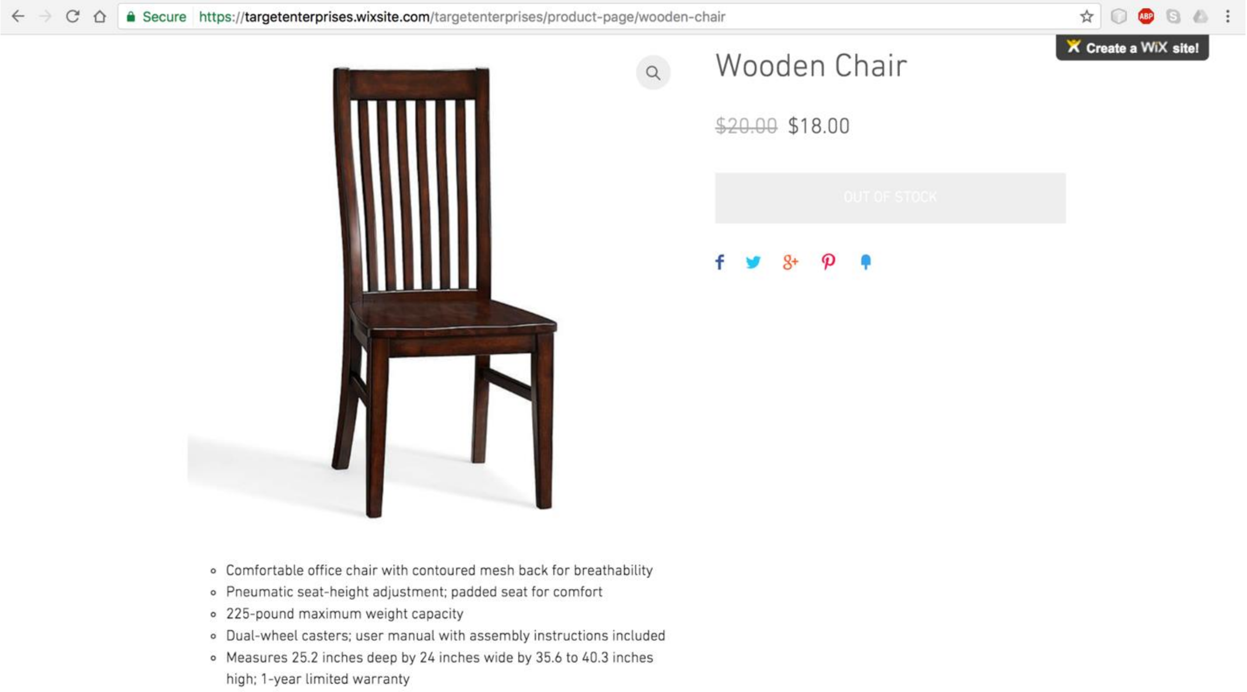
https://targetenterprises.wixsite.com/targetenterprises/departments

500 Terry Francois Street San Francisco
CA 94158 Tel: 123-456-7890
info@mysite.com

After logging in, the clerk would see this page which shows the ongoing sales. He even has an option to log out.

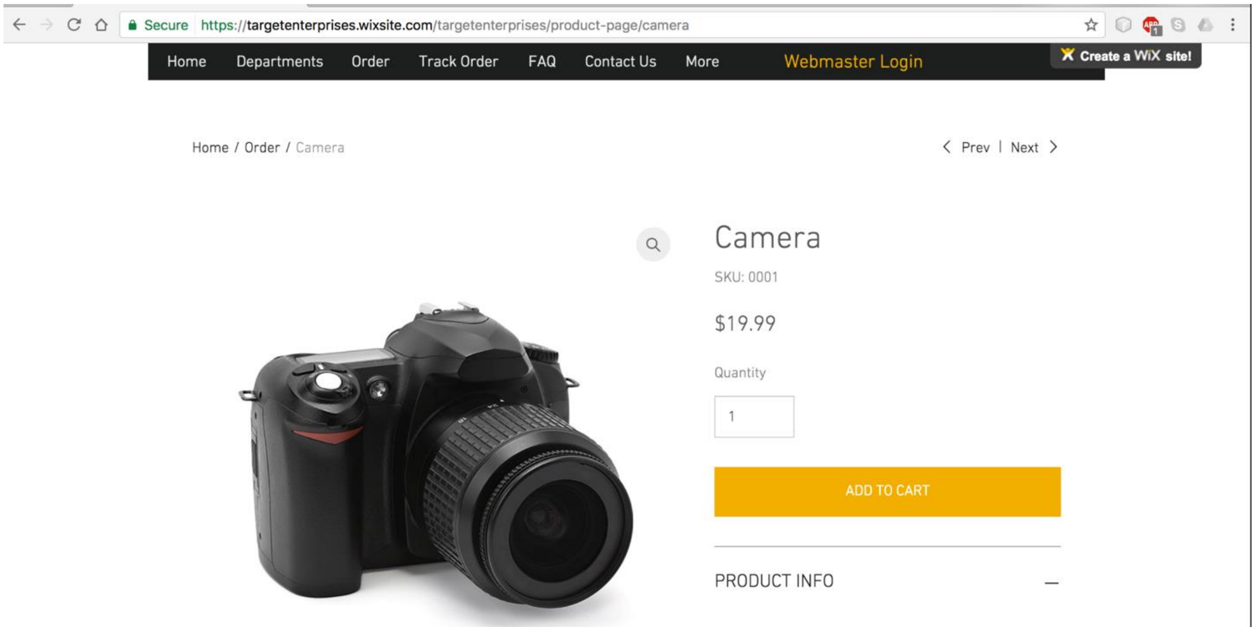


This is the order page where clerk can see various products available on the website. The products have tags like “New arrival” and “New”. New Arrivals would be the latest products which came recently on “Target Enterprises”.


Secure https://targetenterprises.wixsite.com/targetenterprises/product-page/wooden-chair
Create a WiX site!
Wooden Chair
~~$20.00~~ $18.00
OUT OF STOCK

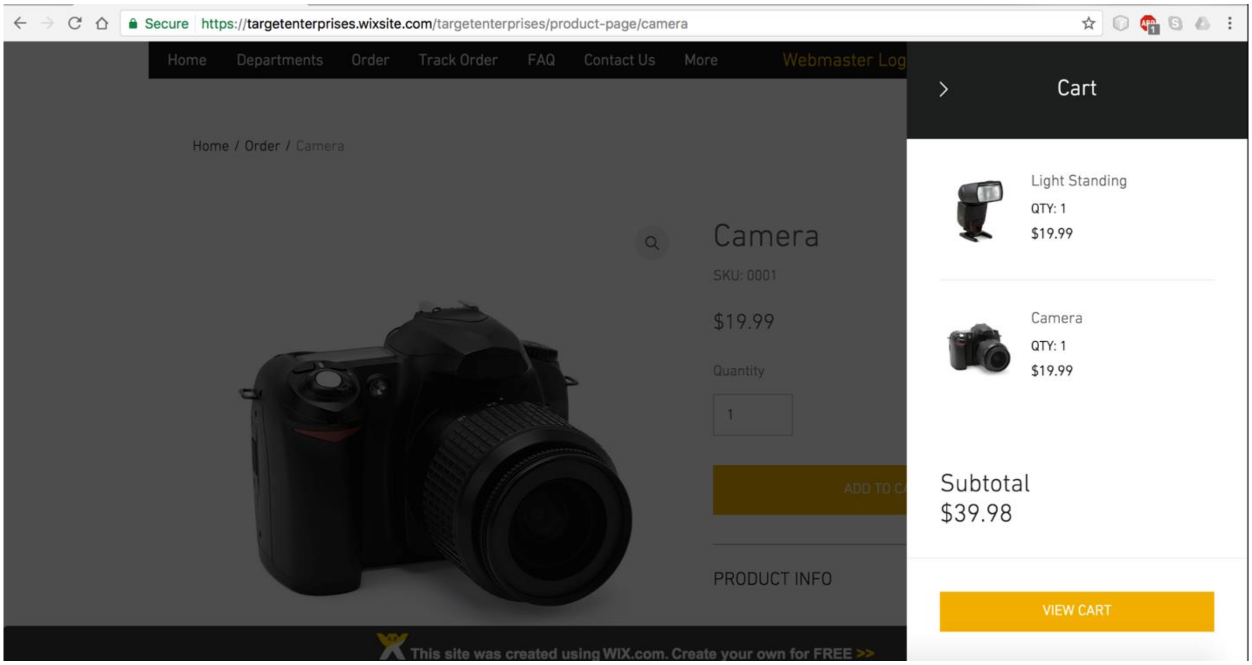
- Comfortable office chair with contoured mesh back for breathability
- Pneumatic seat-height adjustment; padded seat for comfort
- 225-pound maximum weight capacity
- Dual-wheel casters; user manual with assembly instructions included
- Measures 25.2 inches deep by 24 inches wide by 35.6 to 40.3 inches high; 1-year limited warranty

After clicking on the individual product, clerk would be able to see the price of the product and its description. This would give an insight of the product to the clerk, which would help him buy goods for the customer who wants to purchase.

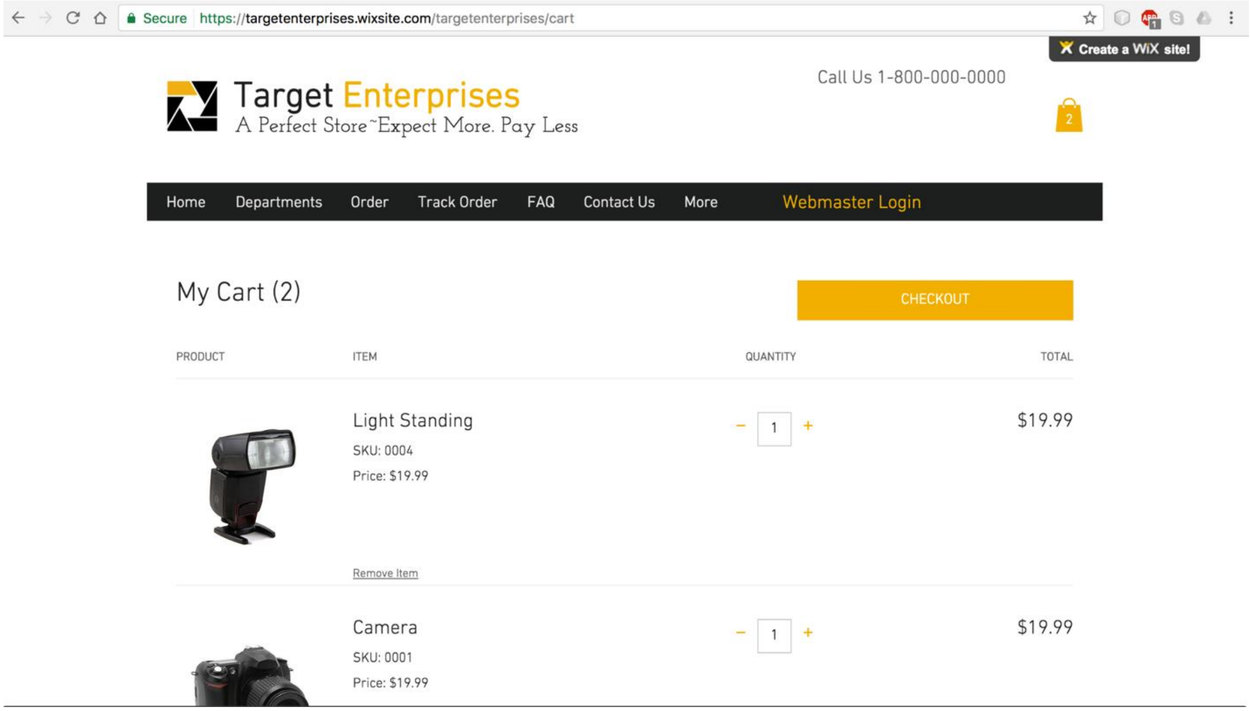


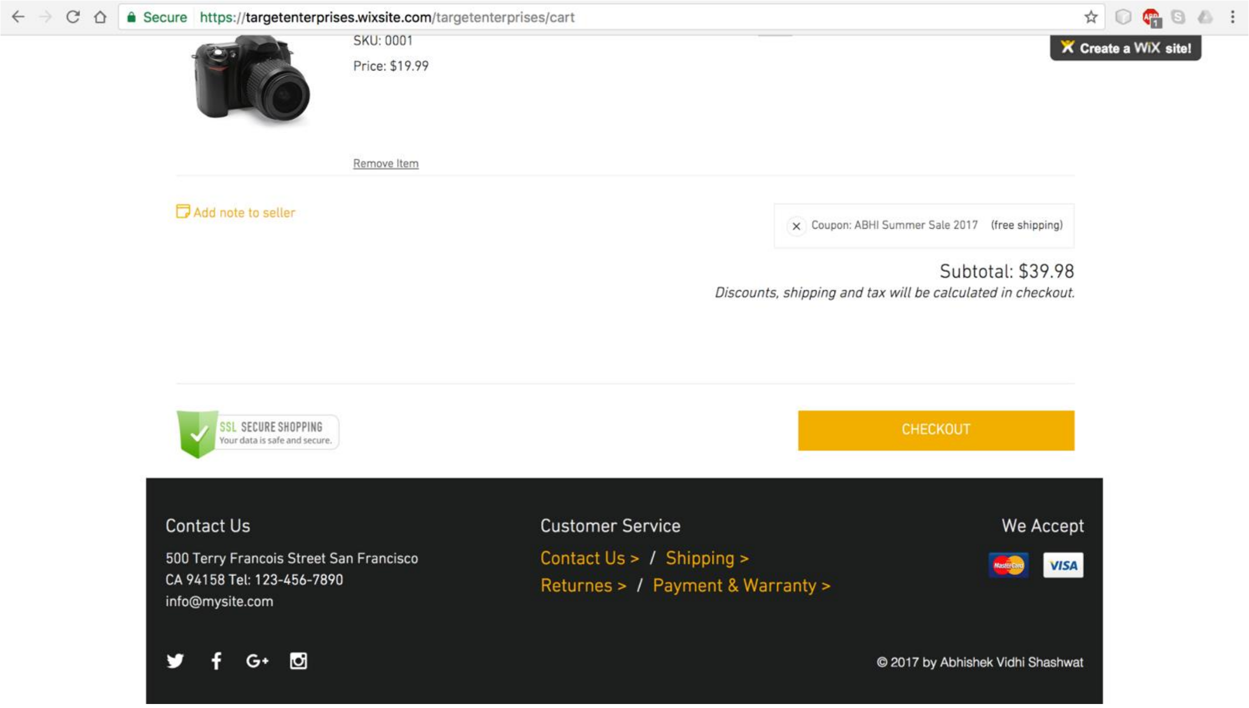
Camera
SKU: 0001
$19.99
Quantity
1

The clerk can add as many products to the shopping cart as the customer wants. He can view the shopping cart later as well.



The clerk can even hover his mouse on the product image which would zoom the product image. This would give clerk a clear idea for the product.





Secure https://targetenterprises.wixsite.com/targetenterprises/cart

Create a WiX site!

SKU: 0001

Price: $19.99

Remove Item

Add note to seller

× Coupon: ABHI Summer Sale 2017 (free shipping)

Subtotal: $39.98

Discounts, shipping and tax will be calculated in checkout.

SSL SECURE SHOPPING
Your data is safe and secure.

CHECKOUT

Contact Us

500 Terry Francois Street San Francisco
CA 94158 Tel: 123-456-7890
info@mysite.com

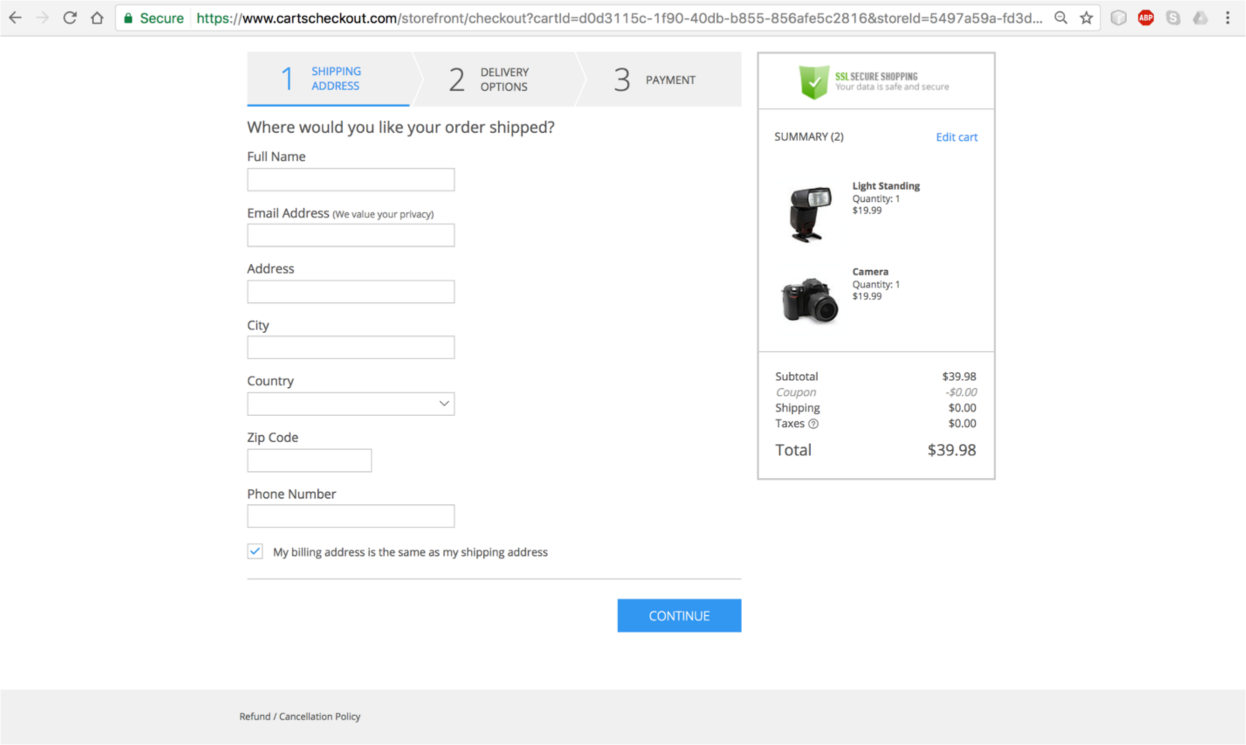
Customer Service

Contact Us > / Shipping >
Returnes > / Payment & Warranty >

We Accept

© 2017 by Abhishek Vidhi Shashwat

The clerk can check out the products he has added in card and proceed for the payments. He can review his order and can still add or remove products he wants. The subtotal amount would also be displayed which will give him an idea on the amount he needs to pay for the products.



Secure https://www.cartscheckout.com/storefront/checkout?cartId=d0d3115c-1f90-40db-b855-856afe5c2816&storeId=5497a59a-fd3d...

1 SHIPPING ADDRESS | 2 DELIVERY OPTIONS | 3 PAYMENT

Where would you like your order shipped?

Full Name

Email Address (We value your privacy)

Address

City

Country

Zip Code

Phone Number

☑ My billing address is the same as my shipping address

CONTINUE

SSL SECURE SHOPPING
Your data is safe and secure

SUMMARY (2) Edit cart

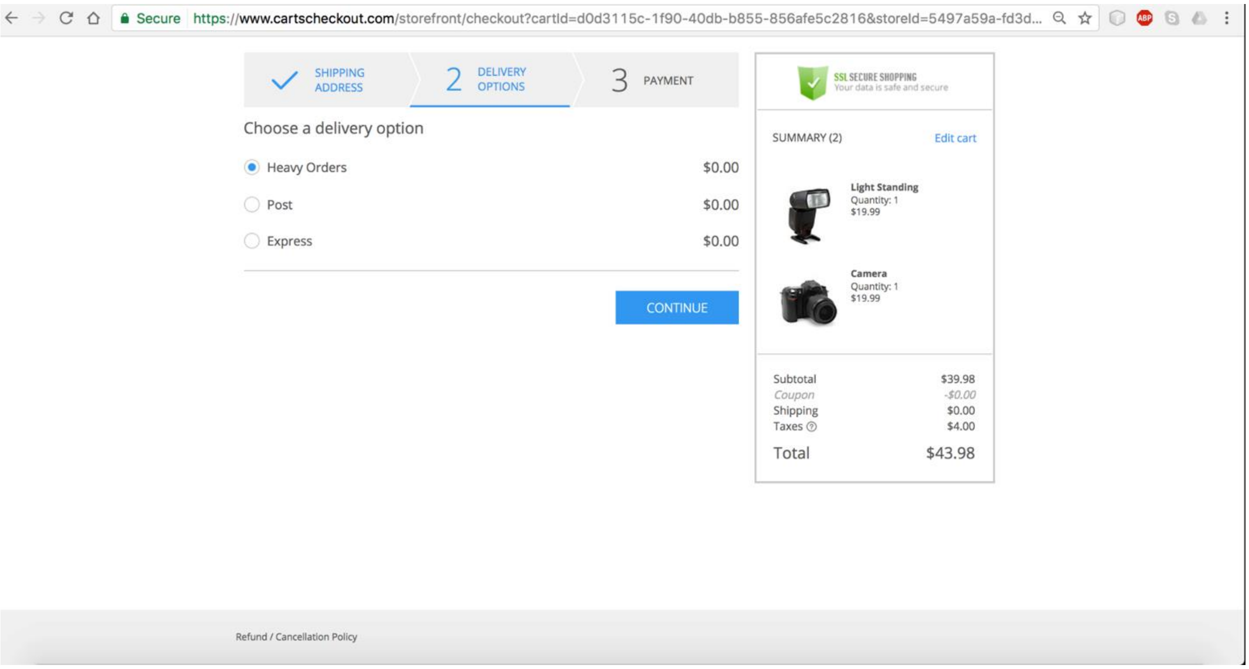
Light Standing
Quantity: 1
$19.99

Camera
Quantity: 1
$19.99

Subtotal	$39.98
Coupon	-$0.00
Shipping	$0.00
Taxes	$0.00
Total	$39.98

Refund / Cancellation Policy

Clerk has to provide the shipping details of the product which includes name of the customer, his email id, and address along with his phone number.



Secure https://www.cartscheckout.com/storefront/checkout?cartId=d0d3115c-1f90-40db-b855-856afe5c2816&storeId=5497a59a-fd3d...

✓ SHIPPING ADDRESS | 2 DELIVERY OPTIONS | 3 PAYMENT

Choose a delivery option

● Heavy Orders	$0.00
○ Post	$0.00
○ Express	$0.00

CONTINUE

SSL SECURE SHOPPING
Your data is safe and secure

SUMMARY (2) Edit cart

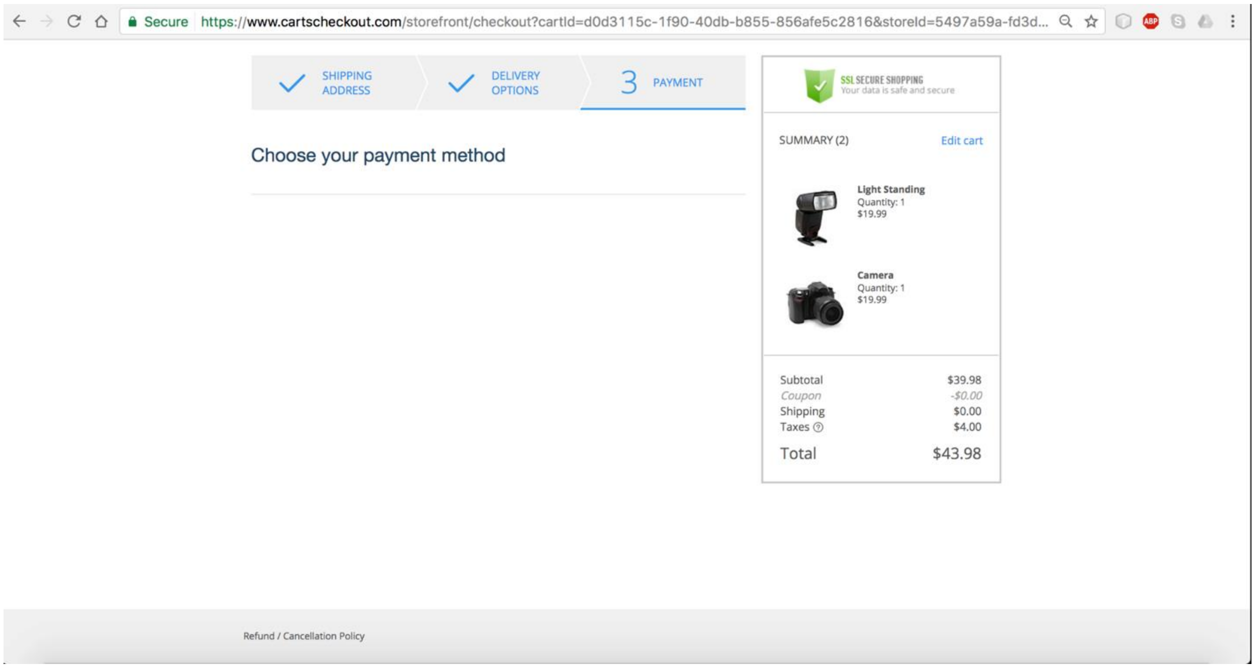
Light Standing
Quantity: 1
$19.99

Camera
Quantity: 1
$19.99

Subtotal	$39.98
Coupon	-$0.00
Shipping	$0.00
Taxes	$4.00
Total	$43.98

Refund / Cancellation Policy

The clerk can choose from a variety of delivery options like Heavy orders, post and express. If the order contains bulky goods like sofa or table, he can select heavy orders. He can select post if it has smaller goods. He also has an option for express delivery in case of emergency orders.



Secure https://www.cartscheckout.com/storefront/checkout?cartId=d0d3115c-1f90-40db-b855-856afe5c2816&storeId=5497a59a-fd3d...

✓ SHIPPING ADDRESS ✓ DELIVERY OPTIONS 3 PAYMENT

Choose your payment method

SSL SECURE SHOPPING
Your data is safe and secure

SUMMARY (2) Edit cart

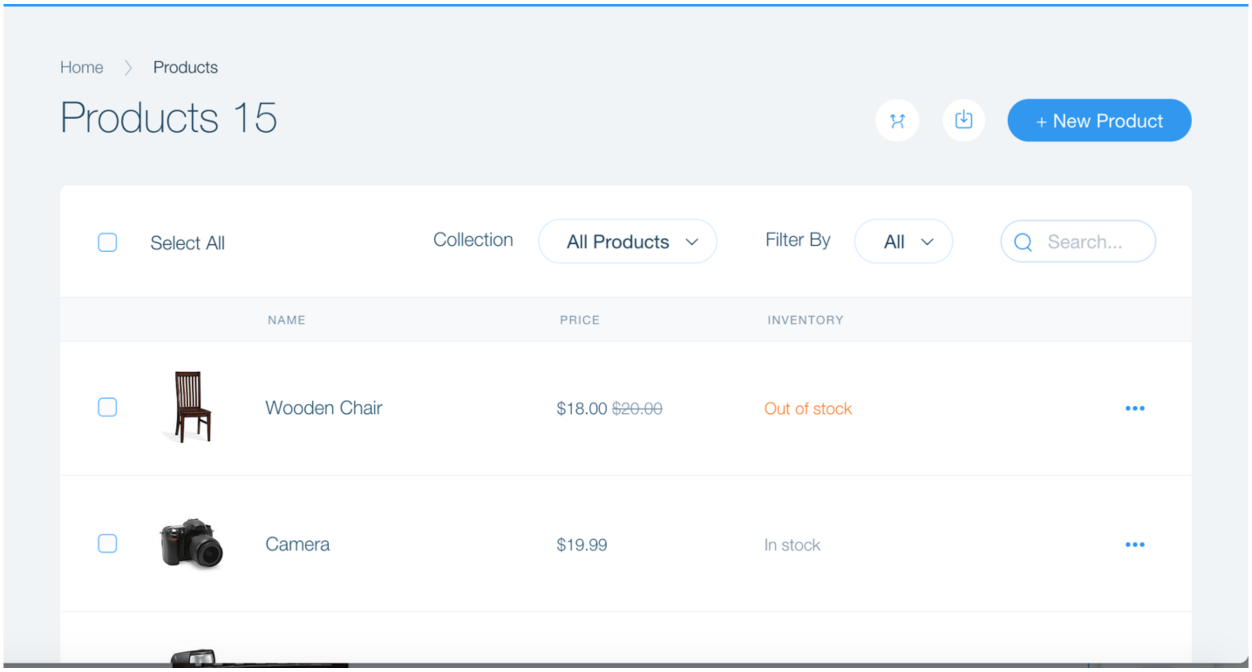
Light Standing
Quantity: 1
$19.99

Camera
Quantity: 1
$19.99

Subtotal	$39.98
Coupon	-$0.00
Shipping	$0.00
Taxes	$4.00
Total	$43.98

Refund / Cancellation Policy

This page would ask the clerk payment methods. He can choose the payment method he wants.



Home > Products

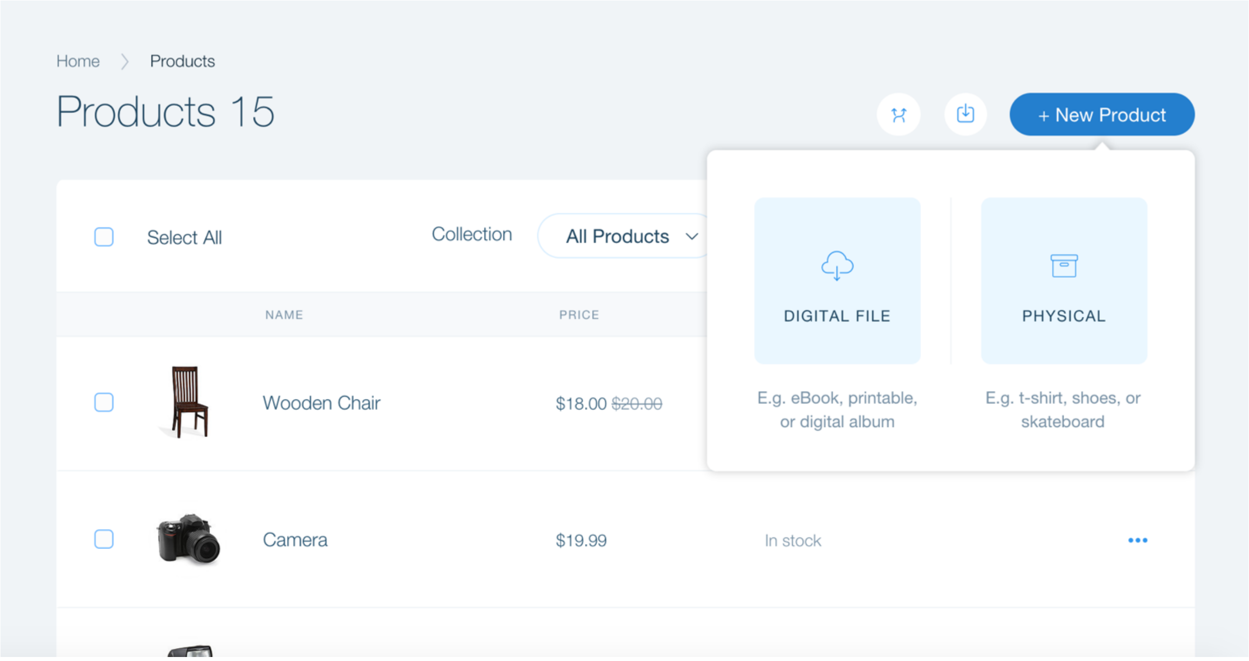
Products 15

+ New Product

□ Select All Collection All Products Filter By All Search...

		NAME	PRICE	INVENTORY	
□		Wooden Chair	$18.00 ~~$20.00~~	Out of stock	•••
□		Camera	$19.99	In stock	•••

This is the inventory page of the website where clerk can manage all the products. The clerk can even search the products on this page where he can update price and other details on the website. He can also update whether the product is in stock or not.



Products 15

□ Select All

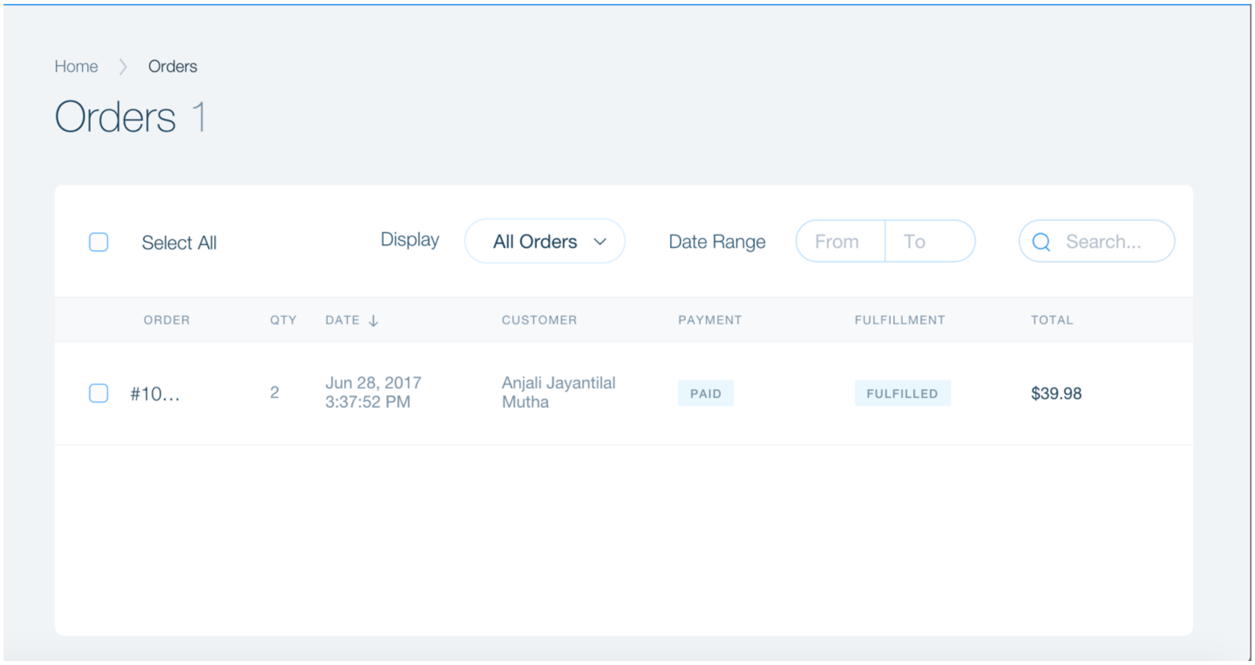
Collection: All Products

	NAME	PRICE	
□	Wooden Chair	$18.00 ~~$20.00~~	
□	Camera	$19.99	In stock

DIGITAL FILE: E.g. eBook, printable, or digital album

PHYSICAL: E.g. t-shirt, shoes, or skateboard

He can add new product on this inventory page. If it is a digital media (eg. eBook, digital album, etc.) that is sold, then clerk would choose digital file. In case of physical product he may choose Physical as an option.



Orders 1

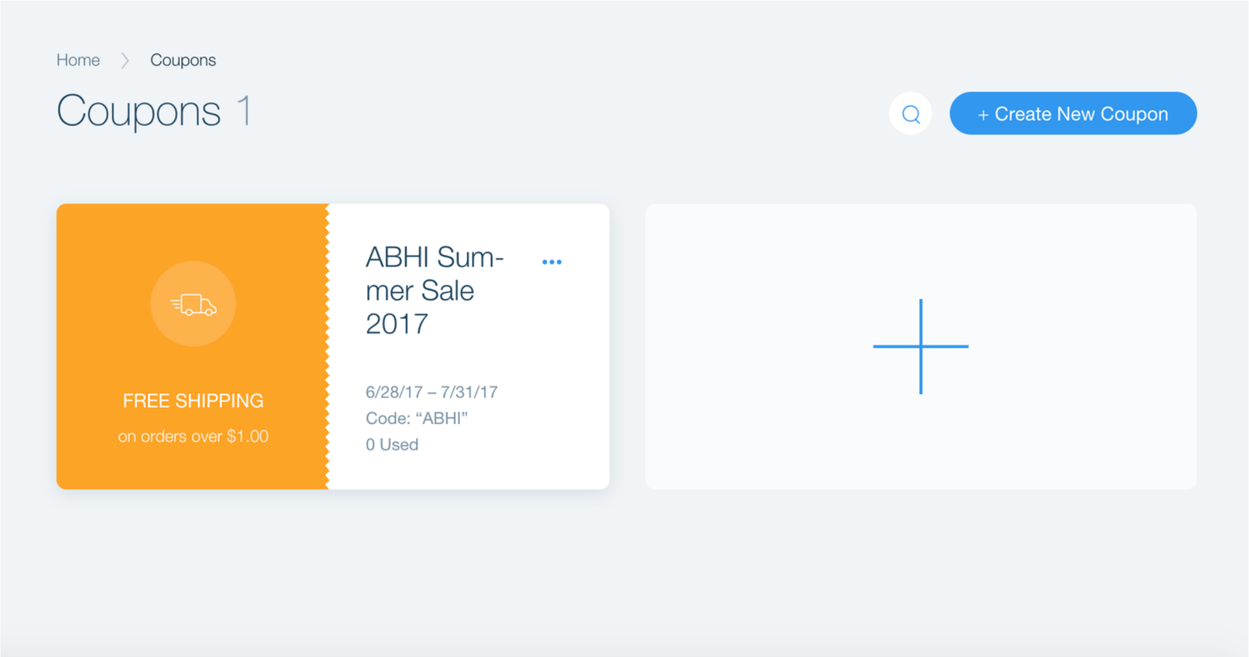
□ Select All

Display: All Orders

Date Range

	ORDER	QTY	DATE ↓	CUSTOMER	PAYMENT	FULFILLMENT	TOTAL
□	#10…	2	Jun 28, 2017 3:37:52 PM	Anjali Jayantilal Mutha	PAID	FULFILLED	$39.98

The clerk can track the orders on this page for the customers. He can check if payment is done or not. He can even check if the product has been delivered or not.



FREE SHIPPING
on orders over $1.00

ABHI Summer Sale 2017

6/28/17 – 7/31/17
Code: "ABHI"
0 Used

The clerk can create new coupons on this page and add on the website. These coupons provide various offers on the "Target Enterprises." website.

FUTURE SCOPE

Target Enterprises is a well-known mail order company in Cape Town. Being approached by an international organization for expansion requires target to aggressively update its operations and way of working. For this it is very important for the company to first improve its information management system. In order to achieve this goal, below is a list of all the scope of development for success in future.

Currently the company does the order processing manually, there is no link between customer details and their purchase history, outdated database system and other technologies: dBase 3 systems etc. The recommendations given below can be implemented in the company database in the future for better, accurate and faster information management.

Implement a BI solution tool:

This will help target automate all the manual operations. This will reduce the probability of error. More management oriented reports can be formed. Data can be put to good use and can be used for monitoring data at micro and macro level.

- **Create a website for online ordering:** This will help maintain all the orders and customer profiles in the database. Also the order details and tracking can be done online. The payments can be accepted online. Thus all the data the 5 departments need will be in sync at one place in the database.

- **Use purchase history of consumers for customized promotion**: By using BI tools the purchase patterns of frequent customers can be tracked thus they can be sent customized promotion offers and thus improving consumer relationship.

- **Start marketing on social media platforms:** Most of the target consumers for online shopping use social media platforms. Thus, promotion on social media can help target more customers. Also by using various analytics tools consumer profiles and demographics can also be known to understand the target consumer group.

- **More components can be added to the website:** More components like order tracking for customers, consumer query, new arrivals, etc. can be added to increase visitor footfall on the website.

TEAM DYNAMICS

Being summer break all the teammates were either travelling or working hence it was difficult to meet according to teammate's convenience. As mentioned in the team strategy during the start of the project each team member was assigned specific tasks based on their knowledge of that topic. And every week each team member would update the other member on the work completed and if there were any difficulties faced. If there was a problem all the members would collectively brainstorm on it to find a solution. All the communications were done via Google Drive, emails, calls and text messages. All the monitoring and the documentation were done on Google Drive. This helped us work cohesively as a team.

As it was a 4 weeks long project that needed extensive technical skills we made a road map to work through the project. Each team member would complete their respective milestone every week and this helped us to work on schedule without deviations. Communication was the key factor required in Target Enterprise project. As ultimately it required compilation of the business needs, proposed solutions, creating a database design flow and implementing it on the web application and database created. Except a few minor challenges and delays the whole project went smoothly. All of us made sure that the problems faced were solved jointly. This helped in beneficial exchange of ideas, doubts or errors.

CHALLENGES FACED

Like any project, we too faced a few challenges while working on Target project. The project required thorough knowledge of problem identification, E-R modelling and SQL queries. We had to contemplate a lot on the business problem and requirement identification. Initially, we gave a very generic idea about the problems and suggested solution. But later it was told that problems and solutions had to be specific. For this we them read a lot about these kinds of businesses and the different technologies they used for data and inventory management. Later we were required to create a database and connect it to a web application. Creating a database was a little challenging and making sure the SQL required execute the required results. But as one of the teammate had a very good knowledge of database creation and implementation and the other 2 teammates worked on SQL queries and creating and implementing the web application. Also as the course was intensive it was difficult to match up with the assignment deadlines and doing the project correctly.

OVERALL EXPERIENCE

The Target Enterprise business problem was implementation of all the concepts learnt in theory in the course in a business prototype. This it helped us gain the working knowledge of database processing fundamentals. Also it helped us improve SQL querying. These 2 are one of the most important skills required for a business analyst. Further, as we had to submit a report every week as an update of our work helped us learn how documentation of a project is done. Correct documentation and presentation of the project is as important as writing the write query and creating a robust database model. It was like documenting the each and every module of the project.

The final report which needed compilation and fine-tuning of the previous project checkpoints helped us look at the project as a whole and suggest improvements. It helped us understand the importance of documentation. Also if there was any error while testing the while project we could easily track back the error using the report created. The presentation needed each team member to explain each phase of the Target project. This improved our presentation skills and made us understand how to deliver the key terms and milestones of the project while talking about the project as a whole.

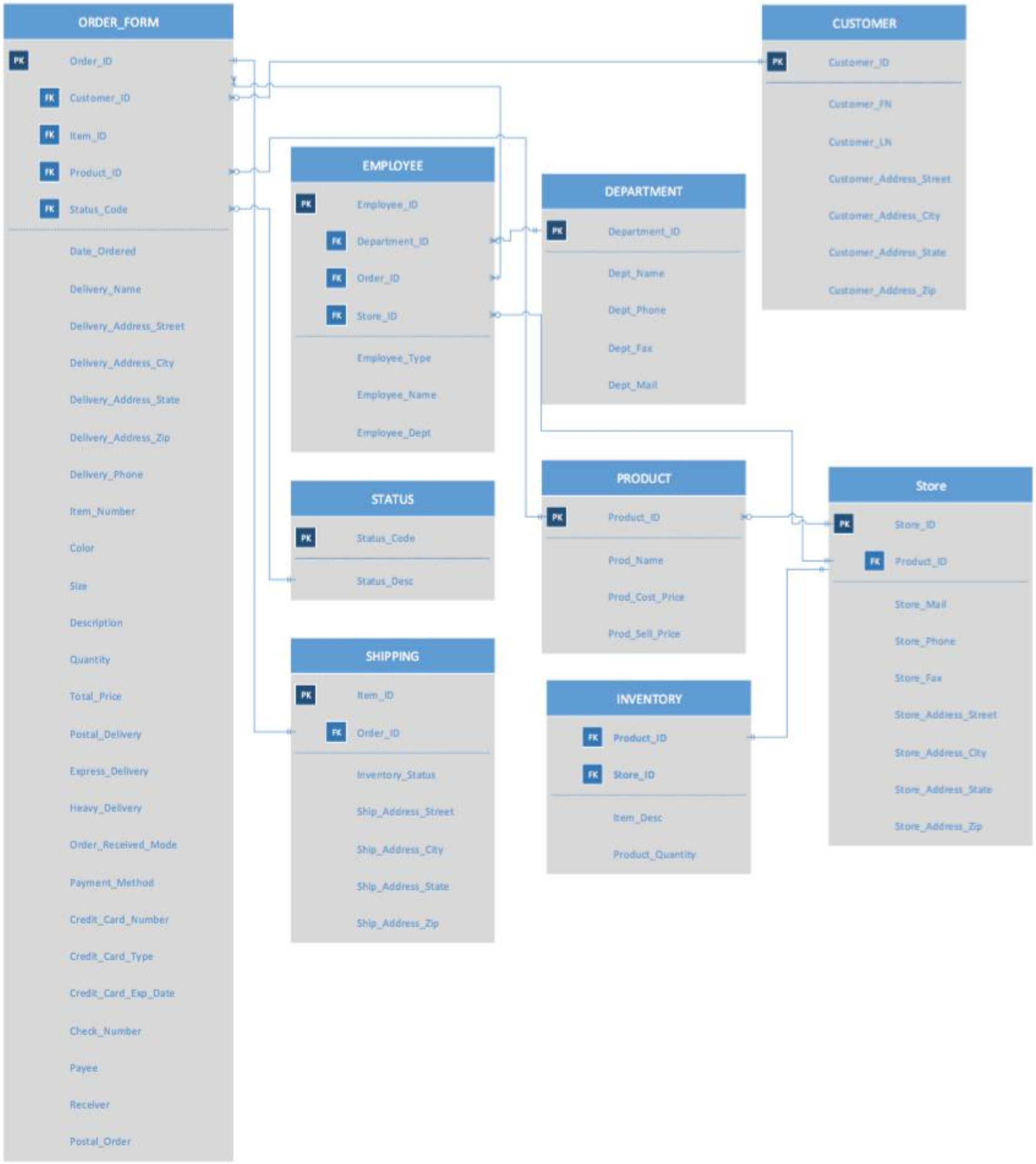
CONCLUSION

The suggested business model will help Target overcome most of the issues faced currently. Not only that it will help them improve the overall working of the company which in turn will save their time and help in business expansion. With the suggested information management system, it will be very easy to create, update and modify the data. Also it will help generate executive summary reports that will help in quick overview of each department and will help concentrate only in areas that need attention. It should overcome all the issues faced by the company right from inventory management, order and dispatch management, data duplication, inadequate reports.

Using the website will let the Target employees access the reports and data seamlessly. This will help target get insights about its business. Thus help them for improvement and expansion of their business in South Africa.

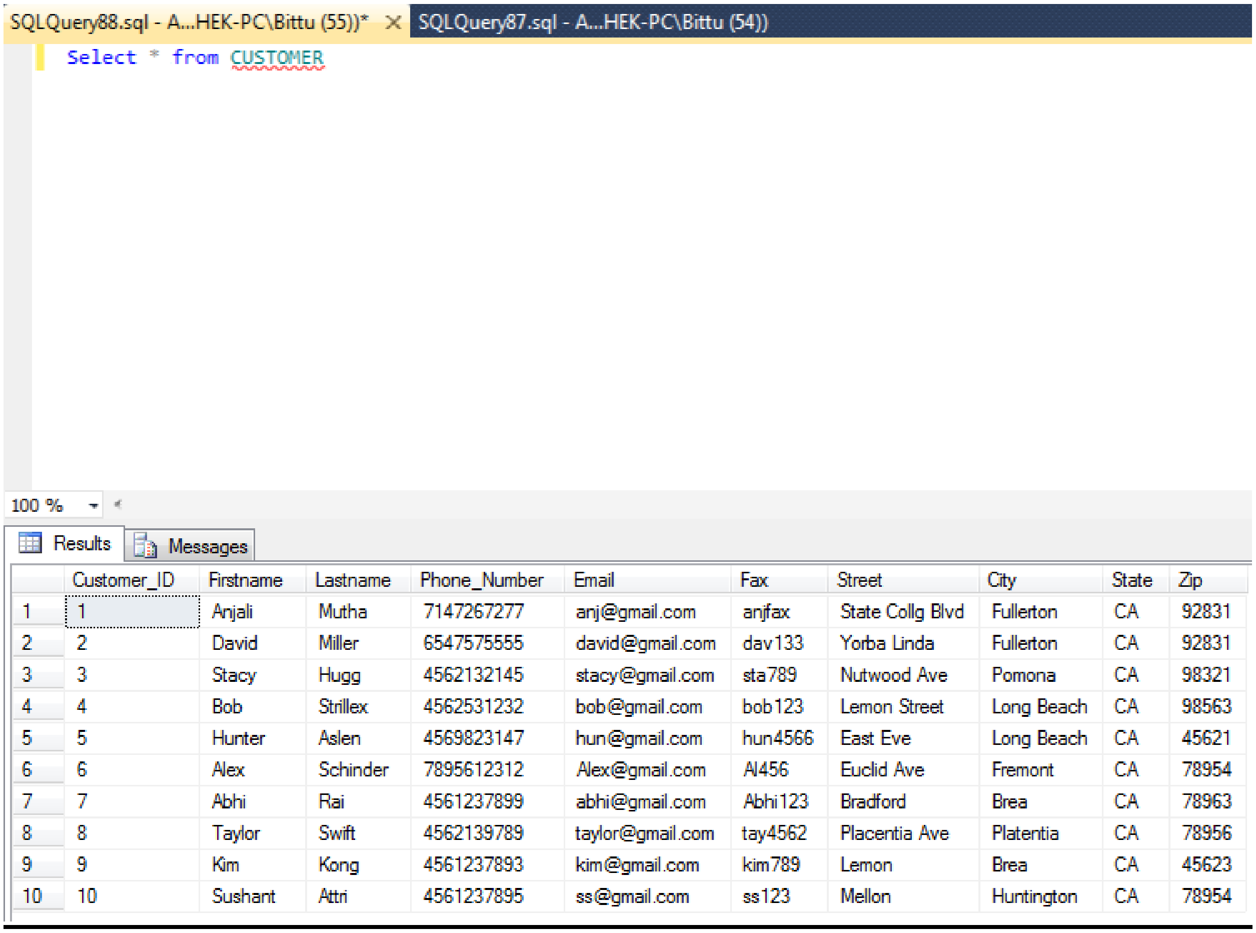
APPENDICES

Old Database Design – Version 1



Screenshots of queries run for various scenarios

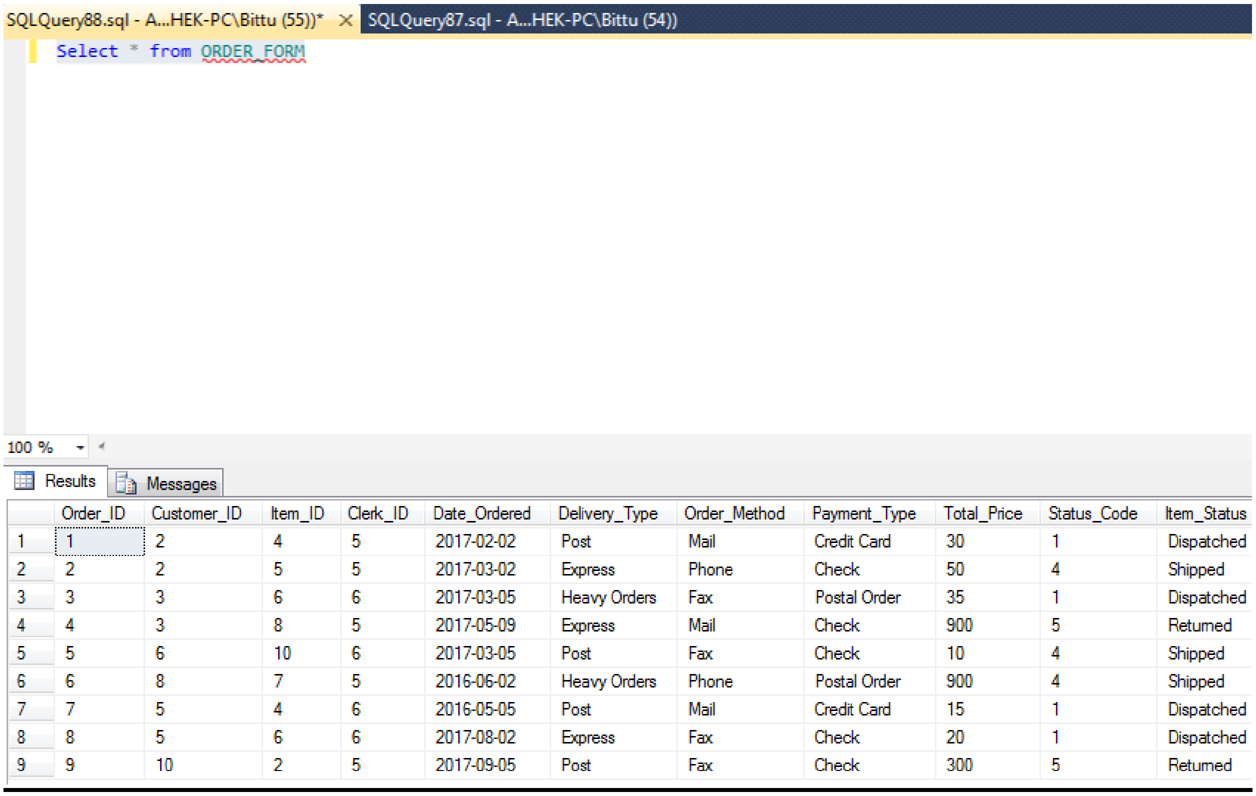
List of Employees


SQLQuery88.sql - A...HEK-PC\Bittu (55))*
SQLQuery87.sql - A...HEK-PC\Bittu (54))
100 %
Results
Messages

```
Select * from CUSTOMER
```

	Customer_ID	Firstname	Lastname	Phone_Number	Email	Fax	Street	City	State	Zip
1	1	Anjali	Mutha	7147267277	anj@gmail.com	anjfax	State Collg Blvd	Fullerton	CA	92831
2	2	David	Miller	6547575555	david@gmail.com	dav133	Yorba Linda	Fullerton	CA	92831
3	3	Stacy	Hugg	4562132145	stacy@gmail.com	sta789	Nutwood Ave	Pomona	CA	98321
4	4	Bob	Strillex	4562531232	bob@gmail.com	bob123	Lemon Street	Long Beach	CA	98563
5	5	Hunter	Aslen	4569823147	hun@gmail.com	hun4566	East Eve	Long Beach	CA	45621
6	6	Alex	Schinder	7895612312	Alex@gmail.com	Al456	Euclid Ave	Fremont	CA	78954
7	7	Abhi	Rai	4561237899	abhi@gmail.com	Abhi123	Bradford	Brea	CA	78963
8	8	Taylor	Swift	4562139789	taylor@gmail.com	tay4562	Placentia Ave	Platentia	CA	78956
9	9	Kim	Kong	4561237893	kim@gmail.com	kim789	Lemon	Brea	CA	45623
10	10	Sushant	Attri	4561237895	ss@gmail.com	ss123	Mellon	Huntington	CA	78954

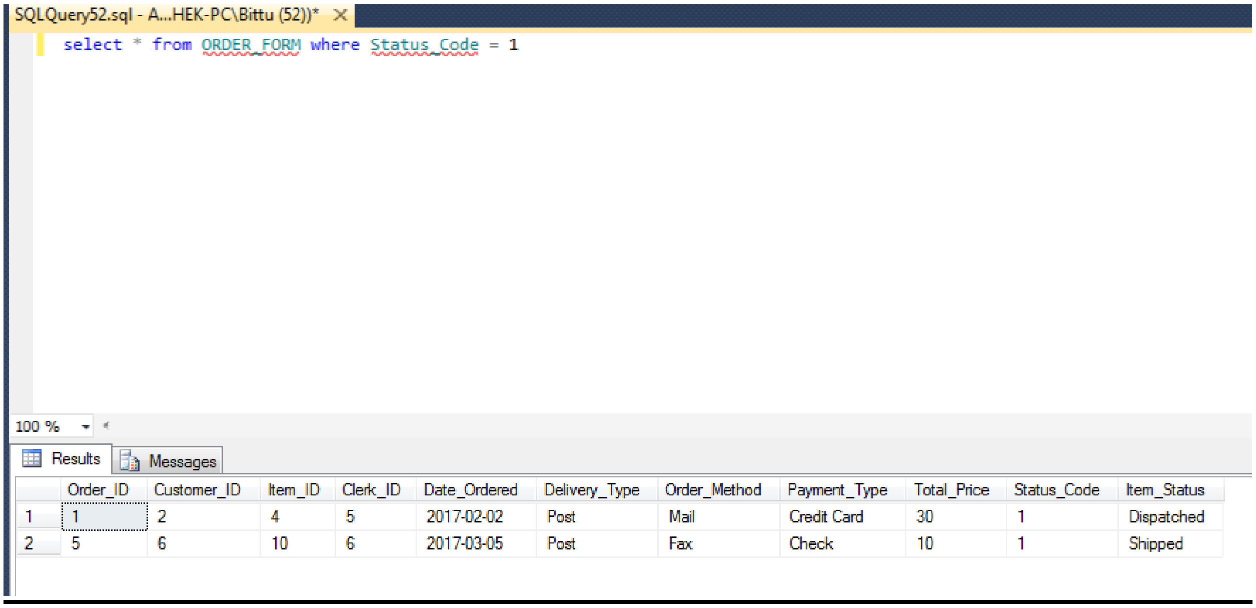
List of Orders in Order Form


SQLQuery88.sql - A...HEK-PC\Bittu (55))*
SQLQuery87.sql - A...HEK-PC\Bittu (54))
100 %
Results
Messages

```
Select * from ORDER_FORM
```

	Order_ID	Customer_ID	Item_ID	Clerk_ID	Date_Ordered	Delivery_Type	Order_Method	Payment_Type	Total_Price	Status_Code	Item_Status
1	1	2	4	5	2017-02-02	Post	Mail	Credit Card	30	1	Dispatched
2	2	2	5	5	2017-03-02	Express	Phone	Check	50	4	Shipped
3	3	3	6	6	2017-03-05	Heavy Orders	Fax	Postal Order	35	1	Dispatched
4	4	3	8	5	2017-05-09	Express	Mail	Check	900	5	Returned
5	5	6	10	6	2017-03-05	Post	Fax	Check	10	4	Shipped
6	6	8	7	5	2016-06-02	Heavy Orders	Phone	Postal Order	900	4	Shipped
7	7	5	4	6	2016-05-05	Post	Mail	Credit Card	15	1	Dispatched
8	8	5	6	6	2017-08-02	Express	Fax	Check	20	1	Dispatched
9	9	10	2	5	2017-09-05	Post	Fax	Check	300	5	Returned

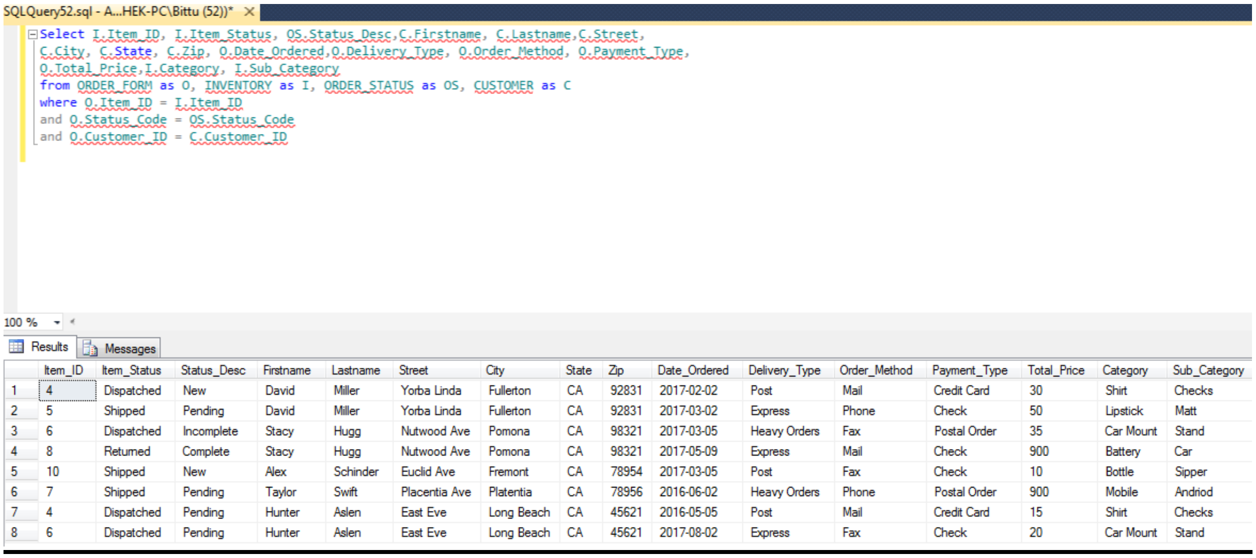
List of new orders-incoming orders



```sql
select * from ORDER_FORM where Status_Code = 1
```

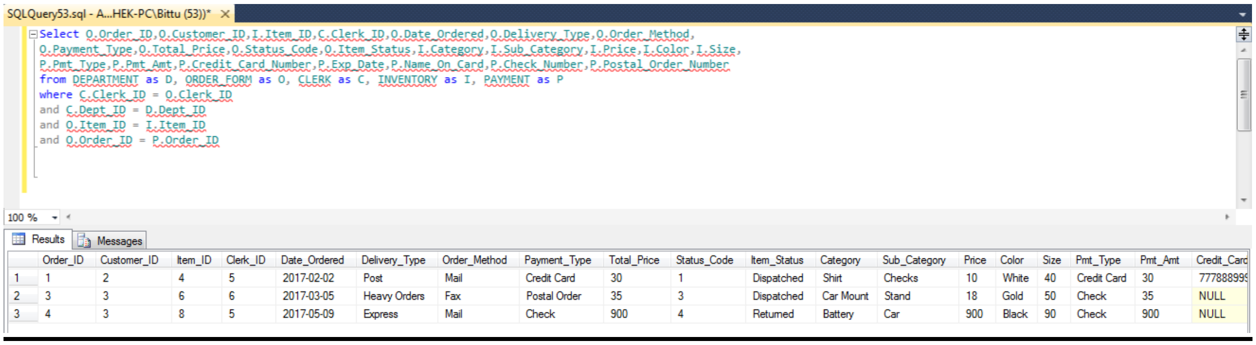
	Order_ID	Customer_ID	Item_ID	Clerk_ID	Date_Ordered	Delivery_Type	Order_Method	Payment_Type	Total_Price	Status_Code	Item_Status
1	1	2	4	5	2017-02-02	Post	Mail	Credit Card	30	1	Dispatched
2	5	6	10	6	2017-03-05	Post	Fax	Check	10	1	Shipped

List of item with payment details



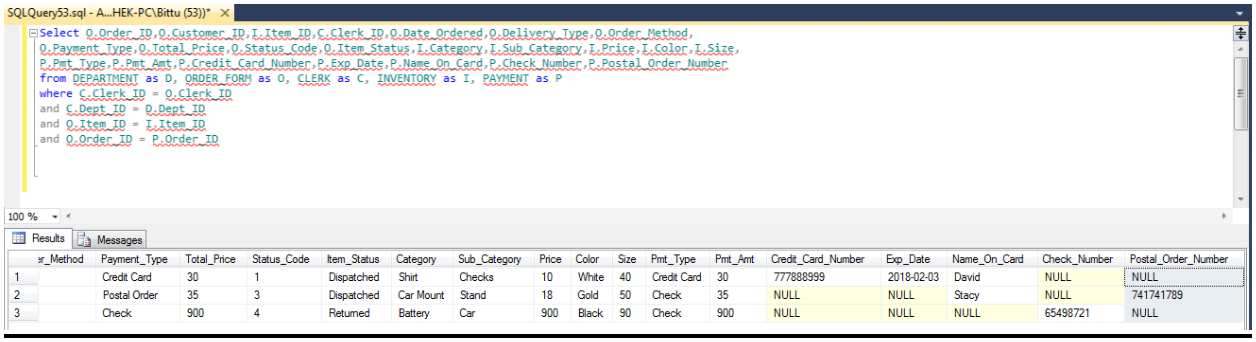
```sql
Select I.Item_ID, I.Item_Status, OS.Status_Desc,C.Firstname, C.Lastname,C.Street,
C.City, C.State, C.Zip, O.Date_Ordered,O.Delivery_Type, O.Order_Method, O.Payment_Type,
O.Total_Price,I.Category, I.Sub_Category
from ORDER_FORM as O, INVENTORY as I, ORDER_STATUS as OS, CUSTOMER as C
where O.Item_ID = I.Item_ID
and O.Status_Code = OS.Status_Code
and O.Customer_ID = C.Customer_ID
```

	Item_ID	Item_Status	Status_Desc	Firstname	Lastname	Street	City	State	Zip	Date_Ordered	Delivery_Type	Order_Method	Payment_Type	Total_Price	Category	Sub_Category
1	4	Dispatched	New	David	Miller	Yorba Linda	Fullerton	CA	92831	2017-02-02	Post	Mail	Credit Card	30	Shirt	Checks
2	5	Shipped	Pending	David	Miller	Yorba Linda	Fullerton	CA	92831	2017-03-02	Express	Phone	Check	50	Lipstick	Matt
3	6	Dispatched	Incomplete	Stacy	Hugg	Nutwood Ave	Pomona	CA	98321	2017-03-05	Heavy Orders	Fax	Postal Order	35	Car Mount	Stand
4	8	Returned	Complete	Stacy	Hugg	Nutwood Ave	Pomona	CA	98321	2017-05-09	Express	Mail	Check	900	Battery	Car
5	10	Shipped	New	Alex	Schinder	Euclid Ave	Fremont	CA	78954	2017-03-05	Post	Fax	Check	10	Bottle	Sipper
6	7	Shipped	Pending	Taylor	Swift	Placentia Ave	Platentia	CA	78956	2016-06-02	Heavy Orders	Phone	Postal Order	900	Mobile	Andriod
7	4	Dispatched	Pending	Hunter	Aslen	East Eve	Long Beach	CA	45621	2016-05-05	Post	Mail	Credit Card	15	Shirt	Checks
8	6	Dispatched	Pending	Hunter	Aslen	East Eve	Long Beach	CA	45621	2017-08-02	Express	Fax	Check	20	Car Mount	Stand



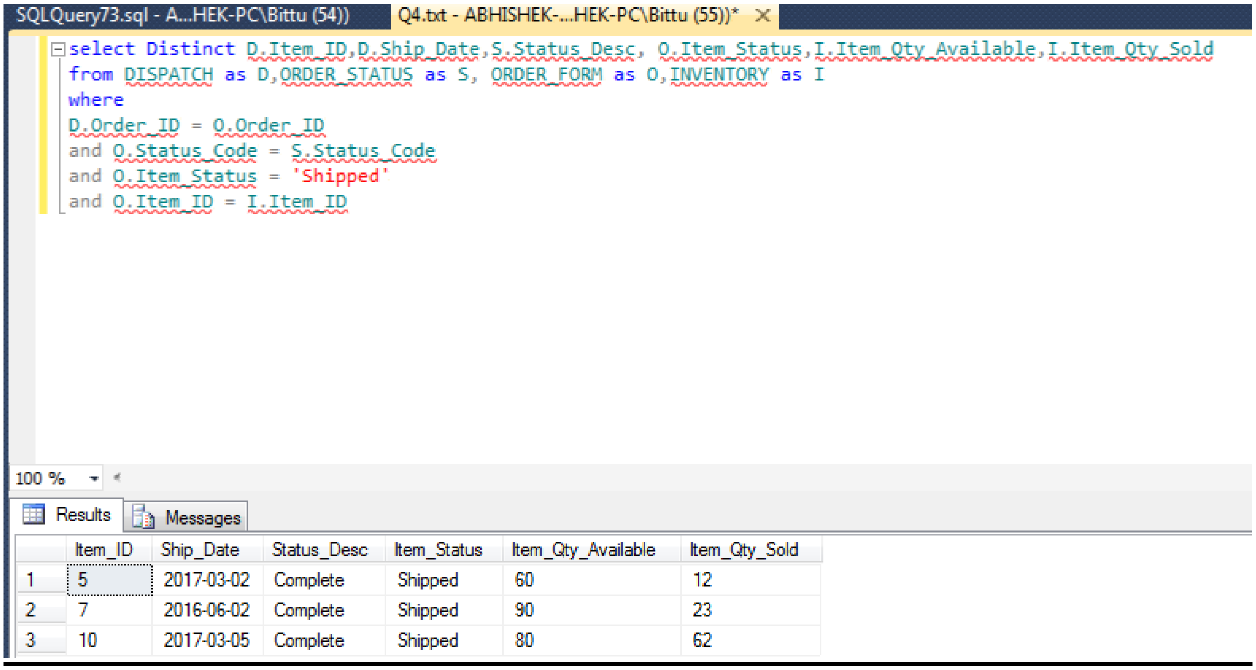
```
Select O.Order_ID,O.Customer_ID,I.Item_ID,C.Clerk_ID,O.Date_Ordered,O.Delivery_Type,O.Order_Method,
O.Payment_Type,O.Total_Price,O.Status_Code,O.Item_Status,I.Category,I.Sub_Category,I.Price,I.Color,I.Size,
P.Pmt_Type,P.Pmt_Amt,P.Credit_Card_Number,P.Exp_Date,P.Name_On_Card,P.Check_Number,P.Postal_Order_Number
from DEPARTMENT as D, ORDER_FORM as O, CLERK as C, INVENTORY as I, PAYMENT as P
where C.Clerk_ID = O.Clerk_ID
and C.Dept_ID = D.Dept_ID
and O.Item_ID = I.Item_ID
and O.Order_ID = P.Order_ID
```

	Order_ID	Customer_ID	Item_ID	Clerk_ID	Date_Ordered	Delivery_Type	Order_Method	Payment_Type	Total_Price	Status_Code	Item_Status	Category	Sub_Category	Price	Color	Size	Pmt_Type	Pmt_Amt	Credit_Card
1	1	2	4	5	2017-02-02	Post	Mail	Credit Card	30	1	Dispatched	Shirt	Checks	10	White	40	Credit Card	30	77788899
2	3	3	6	6	2017-03-05	Heavy Orders	Fax	Postal Order	35	3	Dispatched	Car Mount	Stand	18	Gold	50	Check	35	NULL
3	4	3	8	5	2017-05-09	Express	Mail	Check	900	4	Returned	Battery	Car	900	Black	90	Check	900	NULL



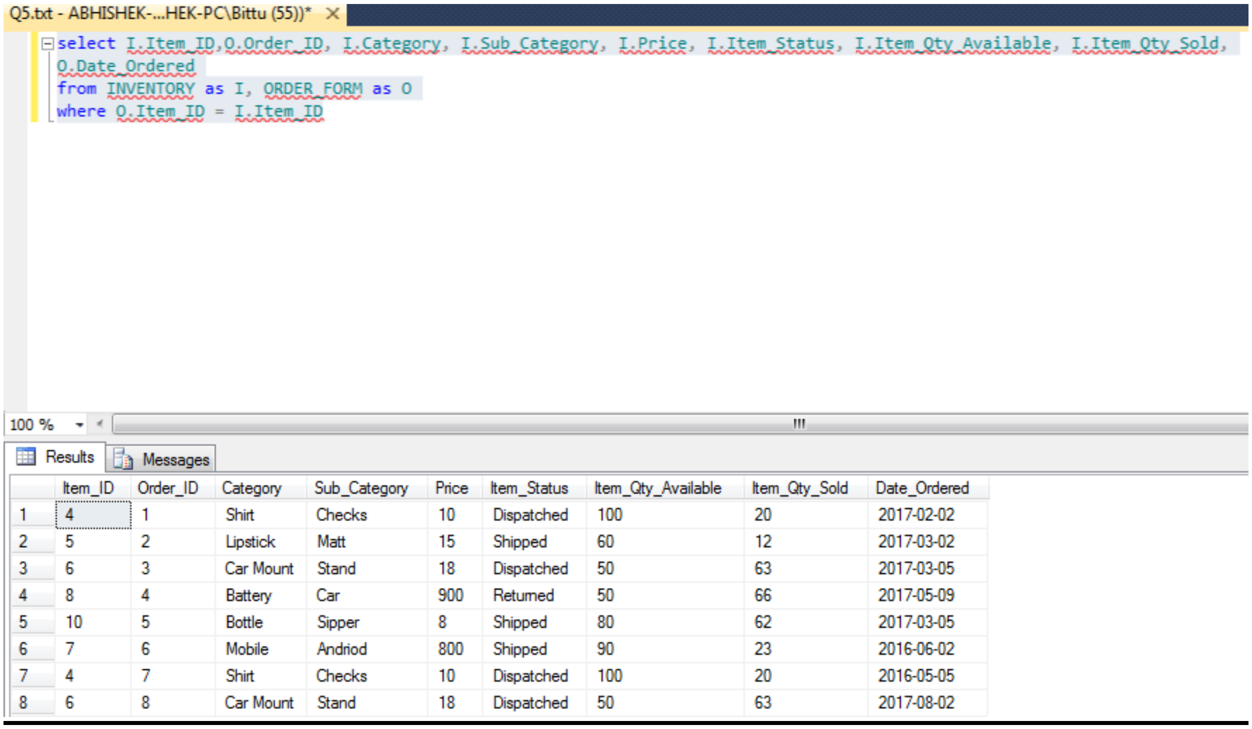
```
Select O.Order_ID,O.Customer_ID,I.Item_ID,C.Clerk_ID,O.Date_Ordered,O.Delivery_Type,O.Order_Method,
O.Payment_Type,O.Total_Price,O.Status_Code,O.Item_Status,I.Category,I.Sub_Category,I.Price,I.Color,I.Size,
P.Pmt_Type,P.Pmt_Amt,P.Credit_Card_Number,P.Exp_Date,P.Name_On_Card,P.Check_Number,P.Postal_Order_Number
from DEPARTMENT as D, ORDER_FORM as O, CLERK as C, INVENTORY as I, PAYMENT as P
where C.Clerk_ID = O.Clerk_ID
and C.Dept_ID = D.Dept_ID
and O.Item_ID = I.Item_ID
and O.Order_ID = P.Order_ID
```

	r_Method	Payment_Type	Total_Price	Status_Code	Item_Status	Category	Sub_Category	Price	Color	Size	Pmt_Type	Pmt_Amt	Credit_Card_Number	Exp_Date	Name_On_Card	Check_Number	Postal_Order_Number
1		Credit Card	30	1	Dispatched	Shirt	Checks	10	White	40	Credit Card	30	777888999	2018-02-03	David	NULL	NULL
2		Postal Order	35	3	Dispatched	Car Mount	Stand	18	Gold	50	Check	35	NULL	NULL	Stacy	NULL	741741789
3		Check	900	4	Returned	Battery	Car	900	Black	90	Check	900	NULL	NULL	NULL	65498721	NULL


SQLQuery73.sql - A...HEK-PC\Bittu (54))
Q4.txt - ABHISHEK-...HEK-PC\Bittu (55))*
100 %
Results
Messages

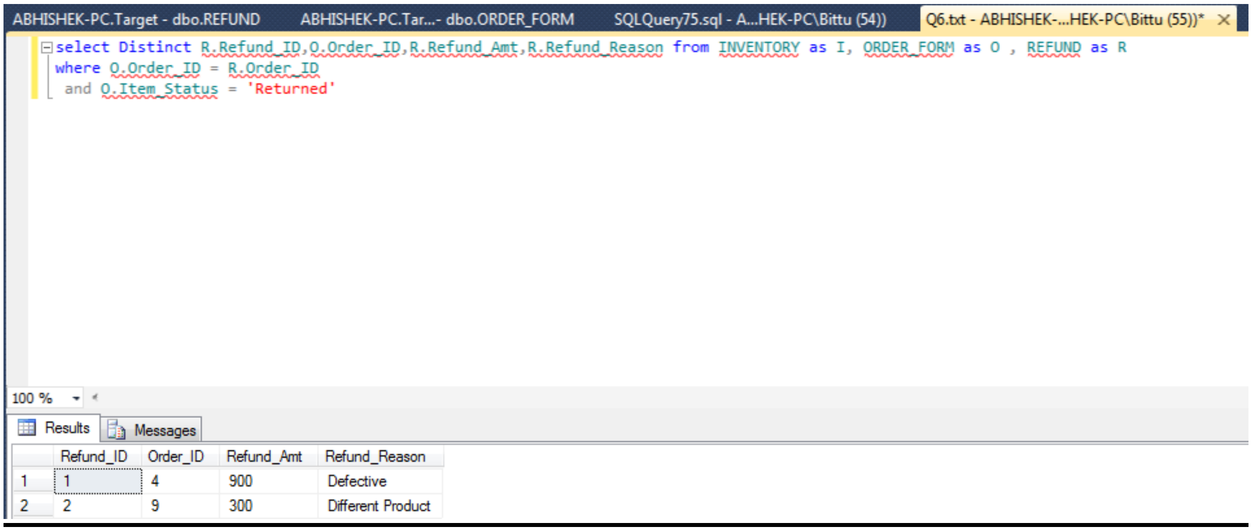
```
select Distinct D.Item_ID,D.Ship_Date,S.Status_Desc, O.Item_Status,I.Item_Qty_Available,I.Item_Qty_Sold
from DISPATCH as D,ORDER_STATUS as S, ORDER_FORM as O,INVENTORY as I
where
D.Order_ID = O.Order_ID
and O.Status_Code = S.Status_Code
and O.Item_Status = 'Shipped'
and O.Item_ID = I.Item_ID
```

	Item_ID	Ship_Date	Status_Desc	Item_Status	Item_Qty_Available	Item_Qty_Sold
1	5	2017-03-02	Complete	Shipped	60	12
2	7	2016-06-02	Complete	Shipped	90	23
3	10	2017-03-05	Complete	Shipped	80	62



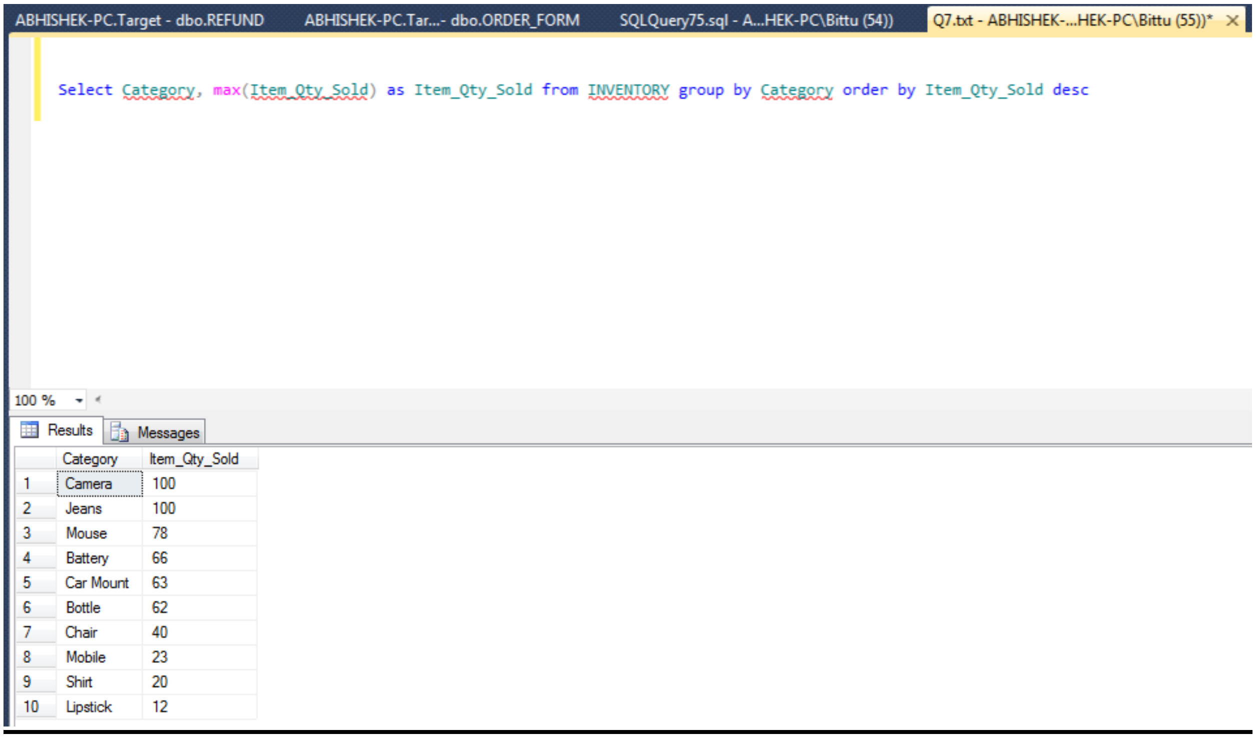
```
select I.Item_ID,O.Order_ID, I.Category, I.Sub_Category, I.Price, I.Item_Status, I.Item_Qty_Available, I.Item_Qty_Sold,
O.Date_Ordered
from INVENTORY as I, ORDER_FORM as O
where O.Item_ID = I.Item_ID
```

	Item_ID	Order_ID	Category	Sub_Category	Price	Item_Status	Item_Qty_Available	Item_Qty_Sold	Date_Ordered
1	4	1	Shirt	Checks	10	Dispatched	100	20	2017-02-02
2	5	2	Lipstick	Matt	15	Shipped	60	12	2017-03-02
3	6	3	Car Mount	Stand	18	Dispatched	50	63	2017-03-05
4	8	4	Battery	Car	900	Returned	50	66	2017-05-09
5	10	5	Bottle	Sipper	8	Shipped	80	62	2017-03-05
6	7	6	Mobile	Andriod	800	Shipped	90	23	2016-06-02
7	4	7	Shirt	Checks	10	Dispatched	100	20	2016-05-05
8	6	8	Car Mount	Stand	18	Dispatched	50	63	2017-08-02


ABHISHEK-PC.Target - dbo.REFUND
ABHISHEK-PC.Tar...- dbo.ORDER_FORM
SQLQuery75.sql - A...HEK-PC\Bittu (54))
Q6.txt - ABHISHEK-...HEK-PC\Bittu (55))*
100 %
Results
Messages

```
select Distinct R.Refund_ID,O.Order_ID,R.Refund_Amt,R.Refund_Reason from INVENTORY as I, ORDER_FORM as O , REFUND as R
where O.Order_ID = R.Order_ID
 and O.Item_Status = 'Returned'
```

	Refund_ID	Order_ID	Refund_Amt	Refund_Reason
1	1	4	900	Defective
2	2	9	300	Different Product


ABHISHEK-PC.Target - dbo.REFUND
ABHISHEK-PC.Tar...- dbo.ORDER_FORM
SQLQuery75.sql - A...HEK-PC\Bittu (54))
Q7.txt - ABHISHEK-...HEK-PC\Bittu (55))*
100 %
Results
Messages

```
Select Category, max(Item_Qty_Sold) as Item_Qty_Sold from INVENTORY group by Category order by Item_Qty_Sold desc
```

	Category	Item_Qty_Sold
1	Camera	100
2	Jeans	100
3	Mouse	78
4	Battery	66
5	Car Mount	63
6	Bottle	62
7	Chair	40
8	Mobile	23
9	Shirt	20
10	Lipstick	12

EXECUTED SQL QUERIES

```
/* Create Table */

CREATE TABLE CUSTOMER (
        Customer_ID    int            NOT NULL, IDENTITY(1,1),
        Firstname      nvarchar(50)   NOT NULL,
        Lastname       nvarchar(50)   NOT NULL,
        Phone_Number   numeric(18, 0) NOT NULL,
        Email          nvarchar(50)   NOT NULL,
        Fax            nvarchar(50)   NOT NULL,
        Street         nvarchar(50)   NOT NULL,
        City           nvarchar(50)   NOT NULL,
        State          nvarchar(50)   NOT NULL,
        Zip            int            NOT NULL,
     Constraint    CustomerPK    PRIMARY KEY (Customer_ID)
     );
```

```
CREATE TABLE ORDER_FORM (
        Order_ID        int         NOT NULL, IDENTITY(1,1),
        Customer_ID     int         NOT NULL,
        Item_ID         int         NOT NULL,
        Clerk_ID        int         NOT NULL,
        Date_Ordered    date        NOT NULL,
        Delivery_Type   nvarchar(50) NOT NULL,
        Order_Method    nvarchar(50) NOT NULL,
        Payment_Type    nvarchar(50) NOT NULL,
        Total_Price     int         NOT NULL,
        Status_Code     int         NOT NULL,
        Item_Status     nvarchar(50) NOT NULL,

   Constraint     Order_FormPK Primary Key (Order_ID),
     Constraint     CustomerFK   Foreign key (Customer_ID)
                         References CUSTOMER (Customer_ID)


                        ON UPDATE NO ACTION
                                  ON DELETE NO ACTION,
     Constraint     ItemFK      Foreign key (Item_ID)
                        References INVENTORY (Item_ID)
                         ON UPDATE NO ACTION
                                  ON DELETE NO ACTION,
     Constraint     ClerkFK     Foreign key (Clerk_ID)
                         References CLERK (Clerk_ID)
                         ON UPDATE NO ACTION
                                  ON DELETE NO ACTION,
     Constraint     Order_StatusFK   Foreign key (Status_Code)
                         References ORDER_STATUS (Status_Code)
                         ON UPDATE NO ACTION
                                  ON DELETE NO ACTION

     );


CREATE TABLE CLERK(
        Clerk_ID    int        NOT NULL, IDENTITY(1,1),
        Dept_ID     int        NOT NULL,
        Store_ID    int        NOT NULL,
        Clerk_Type  nvarchar(50) NOT NULL,
        Clerk_Name  nvarchar(50) NOT NULL,
        Clerk_Dept  nvarchar(50) NOT NULL
     Constraint  ClerkPK      Primary Key (Clerk_ID),
     Constraint      DepartmentFK   Foreign key (Dept_ID)
                        References DEPARTMENT (Dept_ID)
                        ON UPDATE NO ACTION
                                  ON DELETE NO ACTION,
     Constraint      StoreFK        Foreign key (Store_ID)
                        References STORE (Store_ID)
                        ON UPDATE NO ACTION
                                  ON DELETE NO ACTION
     );


CREATE TABLE DEPARTMENT(
        Dept_ID        int            NOT NULL, IDENTITY(1,1),

        Dept_Name       nvarchar(50)    NOT NULL,
        Dept_Phone      numeric(18, 0)  NOT NULL,
        Dept_Fax        nvarchar(50)    NOT NULL,
        Dept_Email      nvarchar(50)    NOT NULL,
     Constraint     DepartmentPK    Primary Key (Dept_ID)
     );


CREATE TABLE DISPATCH(
        Dispatch_ID      int  NOT NULL, IDENTITY(1,1),
        Order_ID         int  NOT NULL,
        Item_ID          int  NOT NULL,
        Customer_ID      int  NOT NULL,
        Ship_Date        date NOT NULL,
     Constraint      DispatchPK    Primary Key (Dispatch_ID),
     Constraint      Order_FormFK   Foreign key (Order_ID)
                        References ORDER_FORM (Order_ID)
                        ON UPDATE NO ACTION

                                  ON DELETE NO ACTION,
     Constraint      InventoryFK   Foreign key (Item_ID)
                        References INVENTORY (Item_ID)
                        ON UPDATE NO ACTION
                                  ON DELETE NO ACTION,
        Constraint      CustomerFK   Foreign key (Customer_ID)
                              References CUSTOMER (Customer_ID)
                              ON UPDATE NO ACTION
                                      ON DELETE NO ACTION
        );


CREATE TABLE INVENTORY(
          Item_ID         int NOT NULL, IDENTITY(1,1),
          Store_ID        int NOT NULL,
          Category        nvarchar(50) NOT NULL,
          Sub_Category    nvarchar(50) NOT NULL,
          Price           int NOT NULL,
          Color           nvarchar(50) NOT NULL,
          Size            nvarchar(50) NOT NULL,
          Item_Status     nvarchar(50) NOT NULL,
          Item_Qty_Available int NOT NULL,
          Item_Qty_Sold     int NOT NULL,
        Constraint       InventoryPK    Primary Key (Item_ID),
        Constraint       StoreFK   Foreign key (Store_ID)
                              References STORE (Store_ID)
                              ON UPDATE NO ACTION
                                      ON DELETE NO ACTION
        );


CREATE TABLE ORDER_STATUS(
          Status_Code int         NOT NULL, IDENTITY(1,1),
          Status_Desc nvarchar(50) NOT NULL,
        Constraint  Order_StatusPK    Primary Key (Status_Code)
        );
CREATE TABLE PAYMENT(
          Pmt_ID            int          NOT NULL, IDENTITY(1,1),
          Order_ID          int          NULL,
          Customer_ID       int          NULL,
          Pmt_Type          nvarchar(50)   NULL,
          Pmt_Amt           int          NULL,
          Credit_Card_Number  numeric(18, 0) NULL,
          Name_On_Card       nvarchar(50)   NULL,
          Exp_Date          date          NULL,
          Check_Number       numeric(18, 0) NULL,
    Postal_Order_Number numeric(18, 0) NULL,
  Constraint       PaymentPK     Primary Key (Pmt_ID),
  Constraint     Order_FormFK   Foreign key (Order_ID)
                     References ORDER_FORM (Order_ID)
                     ON UPDATE NO ACTION
                               ON DELETE NO ACTION,
  Constraint     CustomerFK    Foreign key (Customer_ID)
                     References CUSTOMER (Customer_ID)
                     ON UPDATE NO ACTION
                               ON DELETE NO ACTION,
  Constraint     PaymentFK     Foreign key (Pmt_ID)
                     References PAYMENT (PMT_ID)
                     ON UPDATE NO ACTION
                               ON DELETE NO ACTION
  );


CREATE TABLE REFUND(


    Refund_ID        int         NOT NULL, IDENTITY(1,1),
    Order_ID        int         NOT NULL,
    Customer_ID      int         NOT NULL,
    Pmt_ID          int         NOT NULL,
    Item_ID         int         NOT NULL,
    Refund_Amt       int         NOT NULL,
    Refund_Reason    nvarchar(50) NOT NULL,
  Constraint      RefundPK     Primary Key (Refund_ID),
  Constraint     Order_FormFK   Foreign key (Order_ID)
                     References ORDER_FORM (Order_ID)
                     ON UPDATE NO ACTION
                               ON DELETE NO ACTION,
  Constraint     CustomerFK   Foreign key (Customer_ID)
                     References CUSTOMER (Customer_ID)
                     ON UPDATE NO ACTION
                               ON DELETE NO ACTION,
  Constraint     PaymentFK   Foreign key (Pmt_ID)
                    References PAYMENT (Pmt_ID)
                     ON UPDATE NO ACTION
                               ON DELETE NO ACTION,
  Constraint     InventoryFK   Foreign key (Item_ID)
```

```
                        References INVENTORY (Item_ID)
                        ON UPDATE NO ACTION
                                ON DELETE NO ACTION
    );


CREATE TABLE SHIPPING(
      Shipping_ID     int       NOT NULL, IDENTITY(1,1),
      Order_ID        int       NOT NULL,
      Ship_Street     nvarchar(50) NOT NULL,
      Ship_City       nvarchar(50) NOT NULL,
      Ship_State      nvarchar(50) NOT NULL,
      Ship_Zip        int       NOT NULL,
      Ship_Date       date      NOT NULL,
    Constraint      ShippingPK   Primary Key (Shipping_ID),
    Constraint      Order_FormFK   Foreign key (Order_ID)
                        References ORDER_FORM (Order_ID)
                        ON UPDATE NO ACTION
                                ON DELETE NO ACTION
    );


CREATE TABLE STORE(
      Store_ID        int        NOT NULL, IDENTITY(1,1),
      Item_ID         int        NOT NULL,
      Phone           numeric(18, 0) NOT NULL,
      Fax             nvarchar(50)   NOT NULL,
      Email           nvarchar(50)   NOT NULL,
      Street          nvarchar(50)   NOT NULL,
      City            nvarchar(50)   NOT NULL,
      State           nvarchar(50)   NOT NULL,
      Zip             int        NOT NULL,
    Constraint      StorePK      Primary Key (Store_ID),
    Constraint      InventoryFK   Foreign key (Item_ID)
                        References INVENTORY (Item_ID)
                        ON UPDATE NO ACTION
                                ON DELETE NO ACTION
    );
```

```
/* Insert queries */


Insert into CUSTOMER values
(1,'David','Miller',7472678912,'david@gmail.com','dav133,'Yorba Linda','Fullerton','CA',92831)
Insert into CUSTOMER values
(1,’Sachin’,’Nayak’,7472678912,'david@gmail.com’,’sa123,’Lemon’,’Fullerton','CA',92831)
Insert into CUSTOMER values
(1,’Sushant’,’Attri’,7472678912,'david@gmail.com’,’sus189’,’Euclid’,’Brea’,’CA',92831)
Insert into CUSTOMER values
(1,’Abhishek’,’Rai’,7472678912,'david@gmail.com’,’abh78’,’Brandon’,’Long
Beach’,’CA',92831)
Insert into CUSTOMER values
(1,’Anjie’,’Gupta’,7472678912,'david@gmail.com’,’anj133,’Placentia’,’Fullerton','CA',92831)


Insert into DEPARTMENT(1,'Marketing',4567891236,'market134','market@target.com')
Insert into DEPARTMENT(2,’Purchasing’,7891236123,’pur1134’,’purchase@target.com')
Insert into DEPARTMENT(3,’Order Processing’,7147267277,’order134’,’order@target.com')
Insert into DEPARTMENT(4,’Accounting’,12345678902,’accountt134’,’account@target.com')
Insert into DEPARTMENT(5,’Dispatching’,9876543218,’dispatch134’,’dispatch@target.com')

Insert into ORDER_STATUS(1,'New')
Insert into ORDER_STATUS(2,’Pending’)
Insert into ORDER_STATUS(3,’Incomplete’)
Insert into ORDER_STATUS(4,’Complete’)
Insert into ORDER_STATUS(5,’Cancelled’)


Insert into ORDER_FORM(1,1,2,3,'2017-02-03','Express','Phone','Credit
Card',40,1,'Dispatched')
Insert into ORDER_FORM(2,3,4,5,’2017-05-05’,’Heavy Order’,’Mail’,’Check’,40,1,’Shipped’)
Insert into ORDER_FORM(3,4,5,6,’2017-02-03','Express’,’Fax’,’Credit Card',40,1,'Dispatched')
Insert into ORDER_FORM(4,5,6,7,’2017-08-09’,’Post’,’Fax’,’Postal Order’,40,1,’Returned’)
Insert into ORDER_FORM(1,5,2,3,’2017-02-03','Express','Phone','Credit Card',40,1,’Returned’)


Insert into CLERK(1,2,3,’Dispatch','Balli','Dispatch')
Insert into CLERK(1,2,3,’Marketing’,’David’,’Marketing’)
Insert into CLERK(1,2,3,’Order Processing’,’Anjali’,’Order Processing’)
Insert into CLERK(1,2,3,’Purchasing’,’Sushant’,’Purchasing’)
Insert into CLERK(1,2,3,’Accounting’,’Sachin’,’Accounting’)


Insert into INVENTORY(1,2,'Camera','DSLR',40,'Black',10,'Shipped',100,90)
Insert into INVENTORY(2,3,’Stand Mount’,’Car’,30,’Red’,8,’Returned’,120,80)

Insert into INVENTORY(3,4,’Bottle’,’Sipper’,50,’Peach’,10,'Shipped',150,50)
Insert into INVENTORY(4,5,’Laptop’,’Android’,60,'Black',10,’Returned’,190,30)
Insert into INVENTORY(5,6,’Camera’,’stand’,90,'Black',10,’Dispatched’,160,89)

Insert into DISPATCH(1,1,2,3,'2017-01-03')
Insert into DISPATCH(2,3,4,5,’2017-05-03')
Insert into DISPATCH(3,4,5,6,’2016-01-03')
Insert into DISPATCH(4,3,2,3,’2017-09-02’)
Insert into DISPATCH(5,1,2,3,’2017-01-03')


Insert into PAYMENT(1,2,1,’Check',50,NULL,NULL,NULL,12376578,65768285)
Insert into PAYMENT(2,2,1,’Postal’,60,NULL,NULL,NULL,12346588,65463215)
Insert into PAYMENT(3,2,1,’Check’,70,NULL,NULL,NULL,12346588,65463215)
Insert into PAYMENT(4,2,1,’Check’,80,NULL,NULL,NULL,12346588,65463215)
Insert into PAYMENT(5,2,1,’Check',50,NULL,NULL,NULL,12346588,65463215)


Insert into REFUND(1,2,5,1,2,40,'Defective')
Insert into REFUND(2,4,4,5,5,60,'Different Product’)


Insert into SHIPPING(1,2,'Yorba Linda','Fullerton','CA',92831,'2017-03-12')
Insert into SHIPPING(2,3,’Lemon’,’Fullerton','CA',92831,'2017-03-12')
Insert into SHIPPING(3,4,’Placentia’,’Placentia’,’CA',92831,'2017-03-12')
Insert into SHIPPING(4,4,’Euclid’,’Brea’,’CA',92831,'2017-03-12')
Insert into SHIPPING(5,5,’Brandon’,’Long Beach’,’CA',92831,'2017-03-12')


Insert into STORE(1,2,1234567896,'abc123','Store1@target.com','Yorba

Linda','Fullerton','CA',92831)
Insert into STORE(2,3,6789612345,’we78','Store2@target.com’,’Lemon’,’Brea’,’CA',92841)


/* Queries in Scope of the System  */
```

```
SELECT Order_ID
      ,Customer_ID
      ,Item_ID
      ,Clerk_ID
      ,Date_Ordered
      ,Delivery_Type
      ,Order_Method
      ,Payment_Type
      ,Total_Price
      ,Status_Code
      ,Item_Status
FROM dbo.ORDER_FORM
where Status_Code = 1
```

```
SELECT * FROM ORDER_FORM
```

```
Select I.Item_ID, I.Item_Status, OS.Status_Desc,C.Firstname, C.Lastname,C.Street,
C.City, C.State, C.Zip, O.Date_Ordered,O.Delivery_Type, O.Order_Method, O.Payment_Type,
O.Total_Price,I.Category, I.Sub_Category
from ORDER_FORM as O, INVENTORY as I, ORDER_STATUS as OS, CUSTOMER as C
where O.Item_ID = I.Item_ID
and O.Status_Code = OS.Status_Code
and O.Customer_ID = C.Customer_ID
```

```
Select
O.Order_ID,O.Customer_ID,I.Item_ID,C.Clerk_ID,O.Date_Ordered,O.Delivery_Type,O.Order_
Method,
O.Payment_Type,O.Total_Price,O.Status_Code,O.Item_Status,I.Category,I.Sub_Category,I.Pric
e,I.Color,I.Size,
P.Pmt_Type,P.Pmt_Amt,P.Credit_Card_Number,P.Exp_Date,P.Name_On_Card,P.Check_Numb
er,P.Postal_Order_Number
from DEPARTMENT as D, ORDER_FORM as O, CLERK as C, INVENTORY as I,
PAYMENT as P
where C.Clerk_ID = O.Clerk_ID
and C.Dept_ID = D.Dept_ID
and O.Item_ID = I.Item_ID
and O.Order_ID = P.Order_ID
```

```
select D.Dispatch_ID,D.Order_ID,D.Item_ID,D.Customer_ID,D.Ship_Date,S.Status_Desc,
O.Item_Status from DISPATCH as D,ORDER_STATUS as S, ORDER_FORM as O
where D.Order_ID = O.Order_ID
and O.Status_Code = S.Status_Code
and O.Item_Status = 'Shipped'
select Distinct D.Item_ID,D.Ship_Date,S.Status_Desc,
O.Item_Status,I.Item_Qty_Available,I.Item_Qty_Sold
from DISPATCH as D,ORDER_STATUS as S, ORDER_FORM as O,INVENTORY as I
where
D.Order_ID = O.Order_ID
and O.Status_Code = S.Status_Code
and O.Item_Status = 'Shipped'
and O.Item_ID = I.Item_ID
```

```
select I.Item_ID,O.Order_ID, I.Category, I.Sub_Category, I.Price, I.Item_Status,
I.Item_Qty_Available, I.Item_Qty_Sold,
O.Date_Ordered
from INVENTORY as I, ORDER_FORM as O
where O.Item_ID = I.Item_ID
```

```
select Distinct R.Refund_ID,O.Order_ID,R.Refund_Amt,R.Refund_Reason from INVENTORY
as I, ORDER_FORM as O , REFUND as R
where O.Order_ID = R.Order_ID
and O.Item_Status = 'Returned'
```

```
Select Category, max(Item_Qty_Sold) as Item_Qty_Sold from INVENTORY group by Category
order by Item_Qty_Sold desc
```