Advanced Database Management System Project – Spring 2015

PROJECT REPORT

Online Shopping System

GAP Inc.

In partial fulfillment for the course of

ADVANCED DATABASE MANAGEMENT SYSTEM

NORTHEASTERN UNIVERSITY
BOSTON, MASSACHUSSETTS

SPRING 2015

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Abstract

Gap Inc. is a leading global retailer offering clothing, accessories and personal care products for men, women and children under the Gap, Banana Republic, Old Navy, Athleta and Intermix brands. Gap Inc. products are available for purchase in more than 90 countries worldwide through about 3,300 company-operated stores, almost 400 franchise stores, and e-commerce sites.

The brand focusses on marketing strategies by advertising in newspapers, popular websites and banners across the United States.

The customers have an option of becoming a privileged customer or chose to be a regular customer. The privileged customers are offered various coupons that lure them to attractive deals.

Apart from having several outlets across the United States it has a well-designed online portal that provides the customers a quick view of its product line, which eases the burden of travelling to the store. The customers can place orders and make payment online using their credit/debit card. There are different shipping options with additional costs by means of which the customers can get an expedited delivery.

1.2 Purpose of Database

GAP Inc. is a rapidly developing material brand that has as of late chosen to build its income potential by growing its capacities past the immediate geological vicinity. Until recently, it has focused essentially on the expansion of profits through increased marketing expenditures in local newspapers and advertising on popular social networking websites to generate more 'foot hold' in its retail space.

Based on the revenues generated by these traditional measures, the management team at GAP Inc. has decided to invest in e-commerce capabilities, and is planning to sell some of its products through an online store, which will be available for shipment across the US with various shipping options to meet the customer demands and will completely revolutionize their business strategy.

1.3 Project Goals

With the emergence of e-commerce websites being used by customers to cater to their shopping needs, the management of GAP Inc. realizes that in order to support an emerging business with an online presence, a new and efficient database design will be required. The database needs to address some of the scenarios as described:

(1) Linking the customer details to the customer id for effective storage and retrieval of the respective customer products.

(2) Selection of required shipping option by customers

(3) Keeping a track of privilege customers and sending e-coupons to attract customers to be members.

(4) Taking customer feedbacks for betterment of the store and improvement.

(5) Making online payments in installments.

(6) Displaying current status of the products available by updating in the database.

2. USER ROLES

The database user is a person who has several privileges on the tables and can modify them in a timely manner. The user roles that the GAP Inc. database supports are:

(1) Administrator

The administrator is a person who manages all the enterprise transactions. He has complete privileges on the tables and can modify the employee salary (except his own), designation of employees, add and modify users and privileges and can generate reports for the management team as per the business needs.

The database designer and management team doesn't have all the privileges for any modifications. So admin is the person who is responsible for such modifications and co-ordination as he has all the rights and privileges.

(2) Employee - Manager

Manager will keep a track of the staff under him. He will be responsible to log their attendance and will assign customer complaints to the staff. He will interact with the customer to handle complains escalated by his team. He will also check the order address and forward it to the respective warehouse.

(3) Employee - Accountant

Accountant will co-ordinate with the manager for payment processing and refunds. His task also involves generating financial reports so as to keep track of company's performance and improvement.

(4) Employee - Warehouse

With consent from his manager, staff employee has access to the inventory table to update the quantity of available products so that current details are available in the warehouse which will

minimize duplicity and confusion. He will co-ordinate with the shipping employee and managers to process the shipments from the warehouse.

(5) Employee - Returns/Exchange

He is responsible of handling returns, cancelled and exchanged orders. He can deal with the customer complaints assigned by his manager or can escalate some special complains to him.

(6) Employee - Shipping

In coordination with warehouse employee , he will handle the shipping part once the order is processed. He will change the product claim status in case the shipment is rejected and will report it to the returns employee.

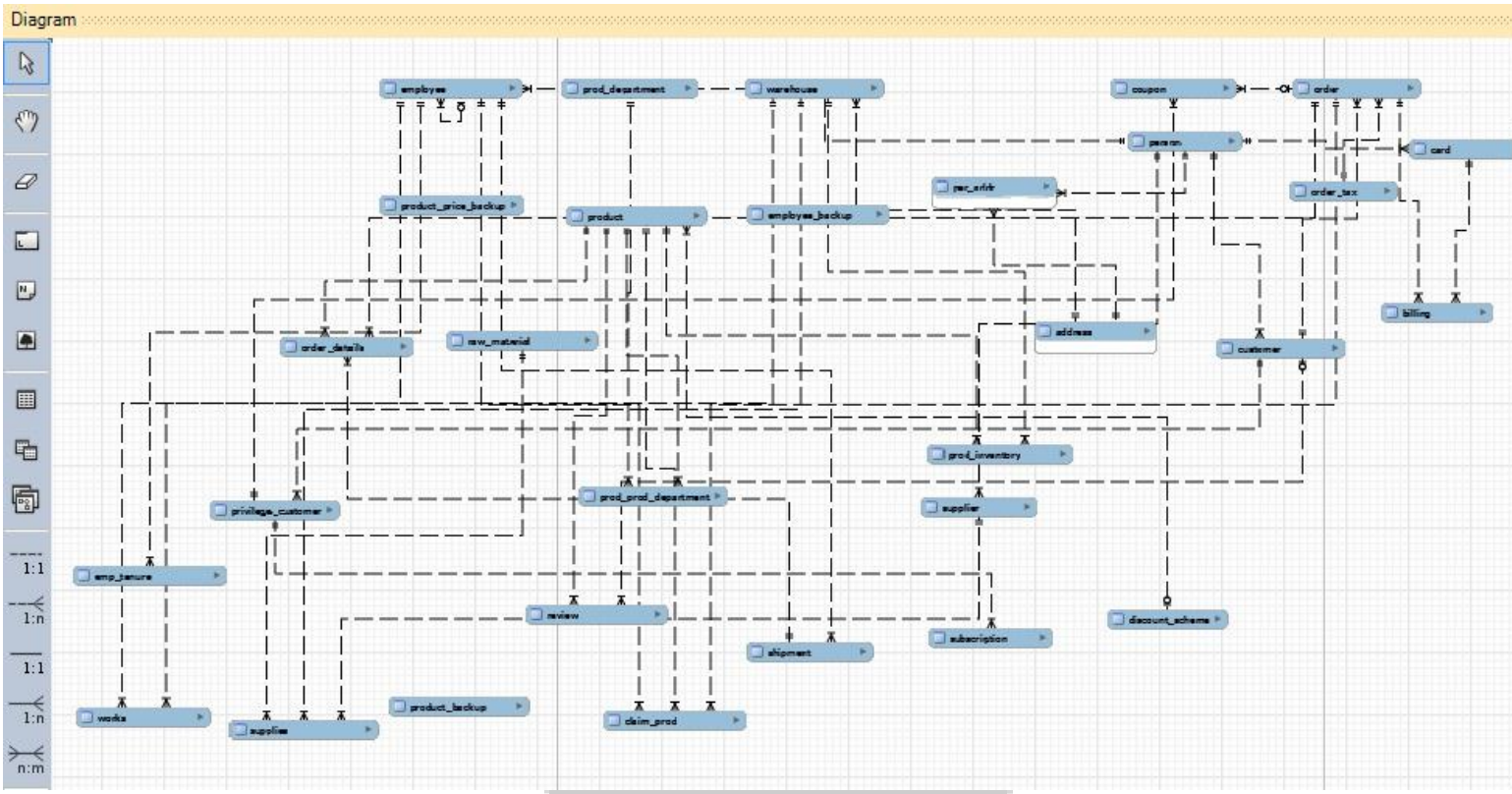
(7) Customer

The customer is a consumer of the products. He will create a login and can choose to be a privileged or regular customer. He has access to the various products and can place orders. He can modify his billing and shipping address, select shipping method, select payment option and claim on the products.

If he is a privileged customer then he can use coupons on the bills and avail discounts and special offers in that period.

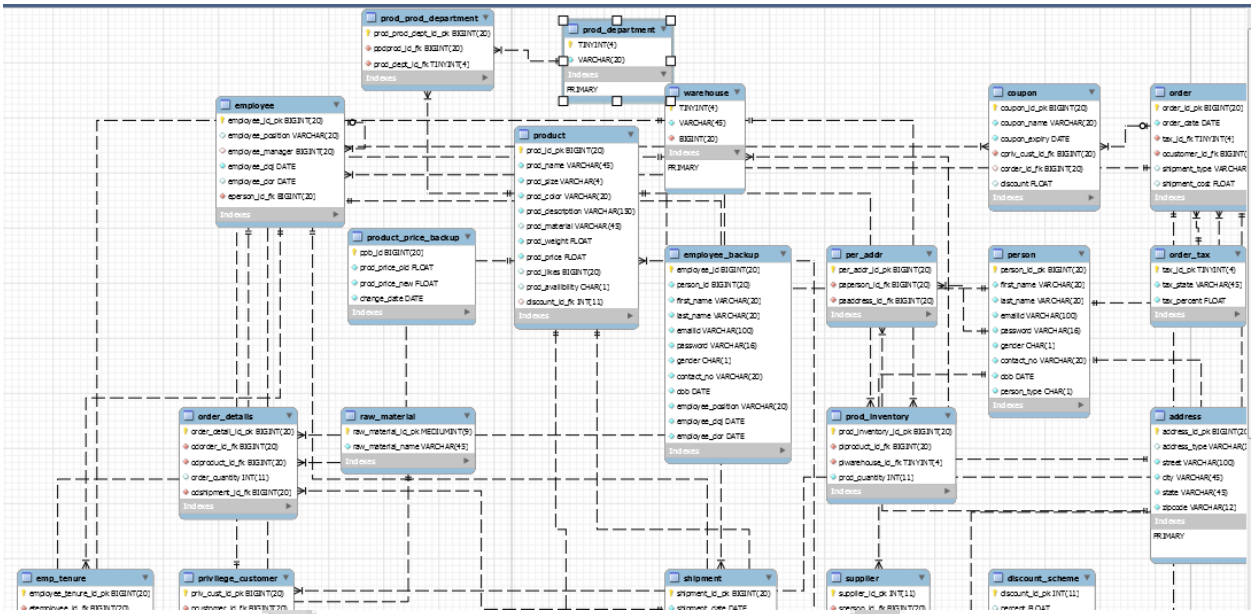
3. CONCEPTUAL MODEL

Below is the conceptual model for my database deisgn:



GAP Store database has in all 30 tables in the model providing a complete picture of an online shopping website used at its best.

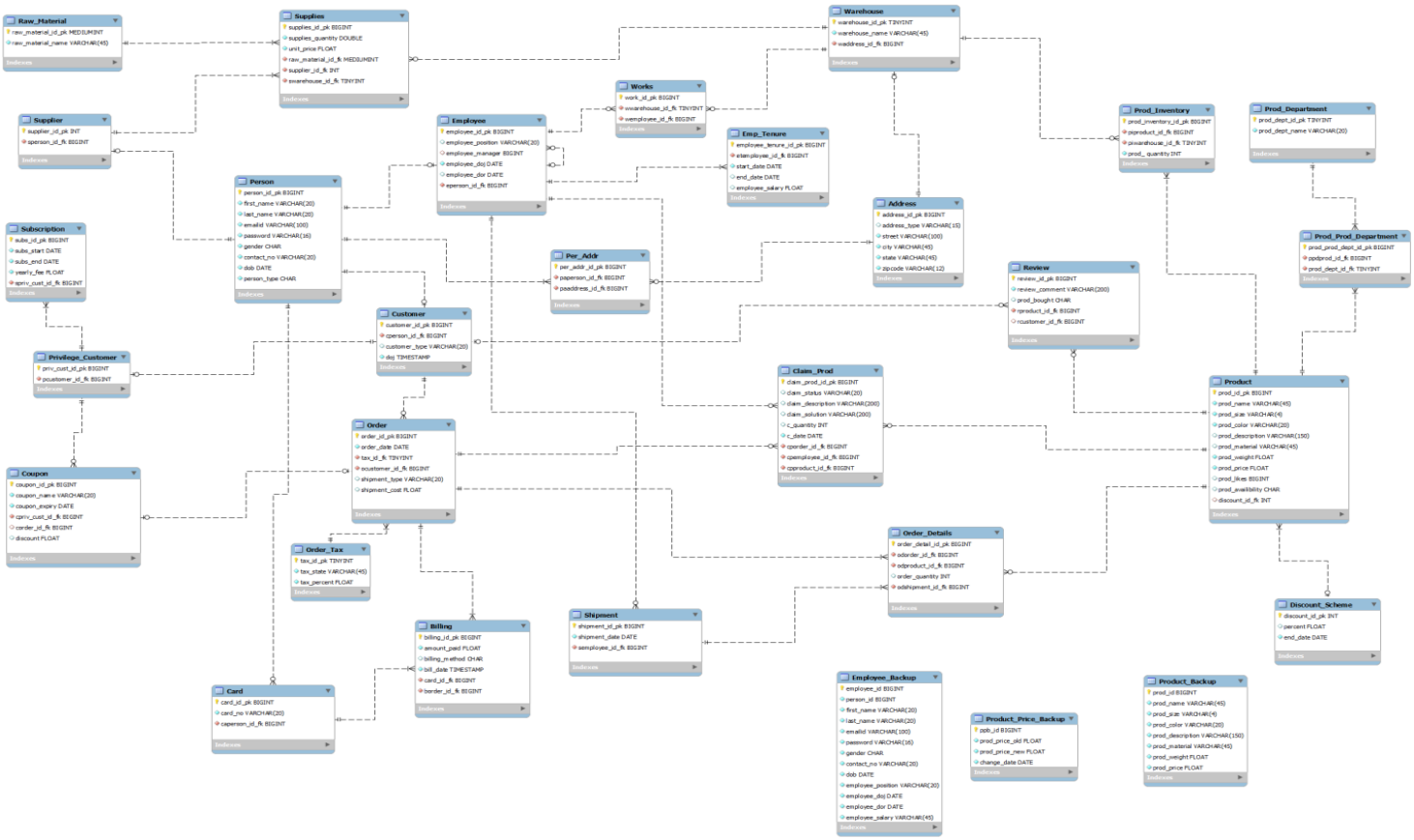
4. LOGICAL DATA MODEL



LogicalGAPstore.mwb

5. PHYSICAL DATA MODEL

Below is the physical model for GAP Inc. case study:



gapStore.mwb

6. VIEWS

1. V_Order_Details: Order Details view by Employee:
   This view shows the all the order details an employee is supposed to see. It shows order ID, order date, customer ID and other customer personal details, shipment type and ID, quantity. This view is a result of Customer, Person, Address, Shipment, Order_Details and Product.

```
1 • select * from v_order_details
```

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

order_id	order_date	customer_id	first_name	last_name	emailid	shipping_address	shipment_type	shipment_
1	2014-04-15	1	nikhil	jangale	nikhil.jangale@gmail.com	24 mahatama, boston, massachussets, 02112	standard	1
1	2014-04-15	1	nikhil	jangale	nikhil.jangale@gmail.com	24 mahatama, boston, massachussets, 02112	standard	1
1	2014-04-15	1	nikhil	jangale	nikhil.jangale@gmail.com	24 mahatama, boston, massachussets, 02112	standard	2
2	2014-04-15	2	hitesh	thavnani	hitesh.thavnani@gmail.com	75 bose, dallas, texas, 09111	standard	3
3	2014-04-17	2	hitesh	thavnani	hitesh.thavnani@gmail.com	75 bose, dallas, texas, 09111	standard	4
3	2014-04-17	2	hitesh	thavnani	hitesh.thavnani@gmail.com	75 bose, dallas, texas, 09111	standard	4

2. V_Product_Details: Product details view as seen by customer:

   This view shows the product code, product name, product size, product color, description, weight, price department name, availability, discount % (if any) and discount expiry. This view is a result of three table: Product, Product Department and the table containing the primary keys of these two tables as foreign key.

```
1 • select * from v_product_details
```

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

product_code	name	size	color	description	material	weight(gms)	price($)	department_name	available	discount(%)	discount_expi
1	polo shirt	m	red	comfortable outing	cotton	40	20	men	y	No Discount!	2015-04-26
2	polo tshirt	l	blue	Super cool!	cotton	50	15	women	y	No Discount!	2015-04-26
3	polo denim	s	grey	summer is on	cotton	60	40	women	y	15	2014-05-01
4	polo casual basic	m	white	casual but smart	cotton	30	10	girl	y	No Discount!	2015-04-26
5	polo scarf	l	black	stylish outer	cotton	20	5	boy	y	No Discount!	2015-04-26
6	blazer	l	black	smart look	cotton	80.5	70	men	y	No Discount!	2015-04-26

3. V_total_prod_quantity: To find the total quantity of each product sold:

```
1 • select * from total_prod_quantity
```

Result Grid | Filter Rows: | Export: | W

prod_id_pk	order_id_pk	Total
1	1	10
1	3	10
1	NULL	20
2	1	5
2	NULL	5
4	1	5
4	3	10
4	NULL	15

4. V_prod_quantity_Warehouse: To find the total quantity of each product sold and the warehouse that supplies it:

```
1 • select * from V_prod_quantity_Warehouse
```

sult Grid | Filter Rows: | Export: | Wrap Ce

prod_id_pk	warehouse_name	Total
1	w1	500
1	w2	300
1	w3	200
1	NULL	1000
2	w1	500
2	w2	300
2	w3	200
2	NULL	1000
3	w1	500

5. V_Manager_Subordinates: To determine manager and his subordinates:

```
1
2 •  select * from  V_Manager_Subordinates
3
```

esult Grid | Filter Rows: | Export:

Manager Name	ManagerID	Subordinates
SELF MANAGED	0	1,5,6,7,10
neha firodiya	1	2,3,4
rucha borle	7	8,9

7. STORED PROCEDURES

1. **P_INSERT_PERSON**: To insert details for an employee/ customer/ person:

We have three types of Person: Employee, Customer and Supplier. This procedure will add the details to the respective tables based on the person type. It also checks for the privilege type of the Person and calculates and inserts the values accordingly in the respective tables.

For eg.,

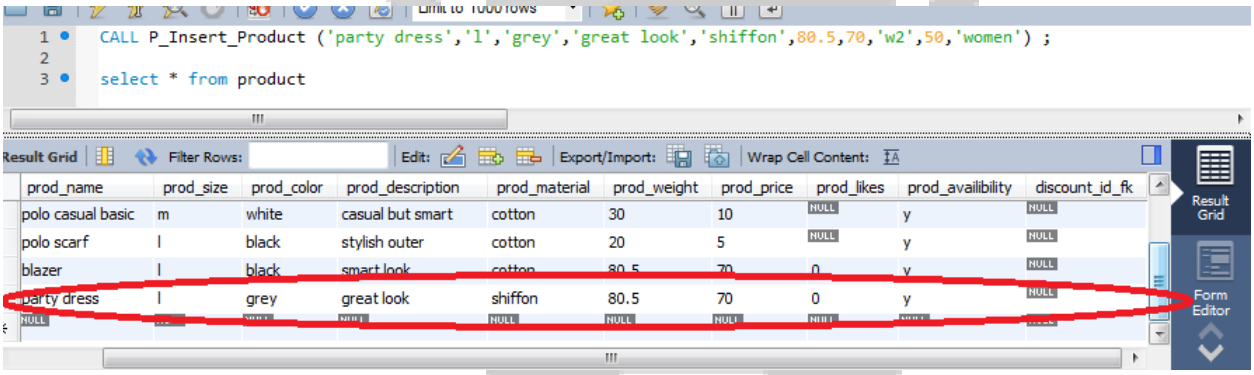
CALL P_Insert_Person(neha, firodiya, neha.firodiya@gmail.com', neha, 'f', '7897538945', '1964-11-01' , 'c', 'regular' , '2014-04-18',null,null,null) ;

This call will add an entry to the tables with the given information in the stored procedure.

2. **P_INSERT_PRODUCT**: To enter product details:

CALL P_Insert_Product ('party dress','l','grey',’great look','shiffon',80.5,70,'w2',50,'women') ;

This call will add a product to the Product table after checking in the warehouse, its availability with all the details



3. **P_ORDER_COST** : Finding cost of a particular Order:

This procedure finds cost of a particular product based on the Product and Order_Details Tables

8. TRIGGERS

1. Trigger **T_Product_Backup:** For taking product backup

This triiger will populate data in the Product_Backup table before deleting any data from the Product table.

2. Trigger **T_employeebackup**: Backing up employee data

This trigger will populate data in the Employee_Backup table and will trigger whenever the data is deleted from the Employee and the associated Person table.

3 Trigger **T_Product_Price_Backup**: For product price backup

This trigger will populate data in the Product_Price_Backup table afterupdating any data in the Product table

2. Trigger **T_Product_Quantity**: For changing in the product quantity

This trigger updates the values in the Prod_Inventory table after inserting the values in Order_Details based on the values in Warehouse table, Person table and Customer tables.

9. BACKUP AND RECOVERY

For backup below script is used to backup my gapStore database

mysqldump -u root -p gapstore > gapstoreBackup.sql

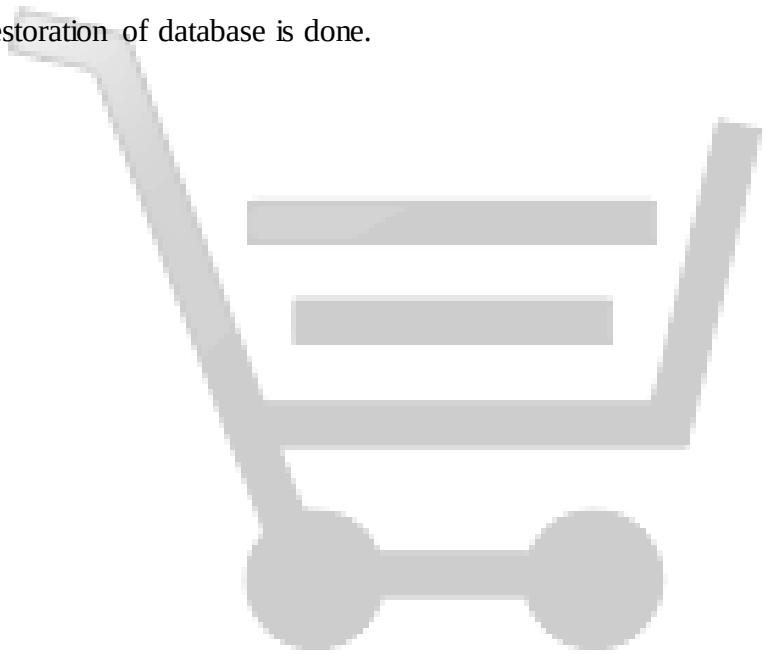
With this script, a backup file of gapStore is created with new name as gapstoreBackup.sql.

Also, backup can be done with the help of the **Data Export** utility available in MySQL.

For restoration, below script is used:

mysql -u root -p gapstoreBackup < gapNew.sql

With this script, restoration of database is done.



10. SECURITY

Database Design and its security is the most important criteria that needs to be considered. It is very necessary to set up privileges in order to restrict access to the people using the database in addition to setting up the passwords and physical security.

There are several ways to secure the database against the vulnerabilities. Below are few ways I have implemented:

1. GRANT option: Granting privileges is an important aspect of database security. I have used various grant options. Listed all in .sql file.
2. If you want to disable network access to your database server, use below command in ini/ cnf file:

   Skip-networking

3. We can delete the test database that is by default created in the MySQL as this database is accessible to all anonymous users. Use below command for dropping the test database:

   DROP database test

4. MySQL database comes with anonymous users with no passwords. You can check this with below command:

   *select * from mysql.user where user="";*

   If it returns something, drop the user using:

   DROP user "";

5. Use passwords for the users created. This is the most basic security aspect everyone must follow so as to secure the database with malicious intrusion.

Granting privileges:

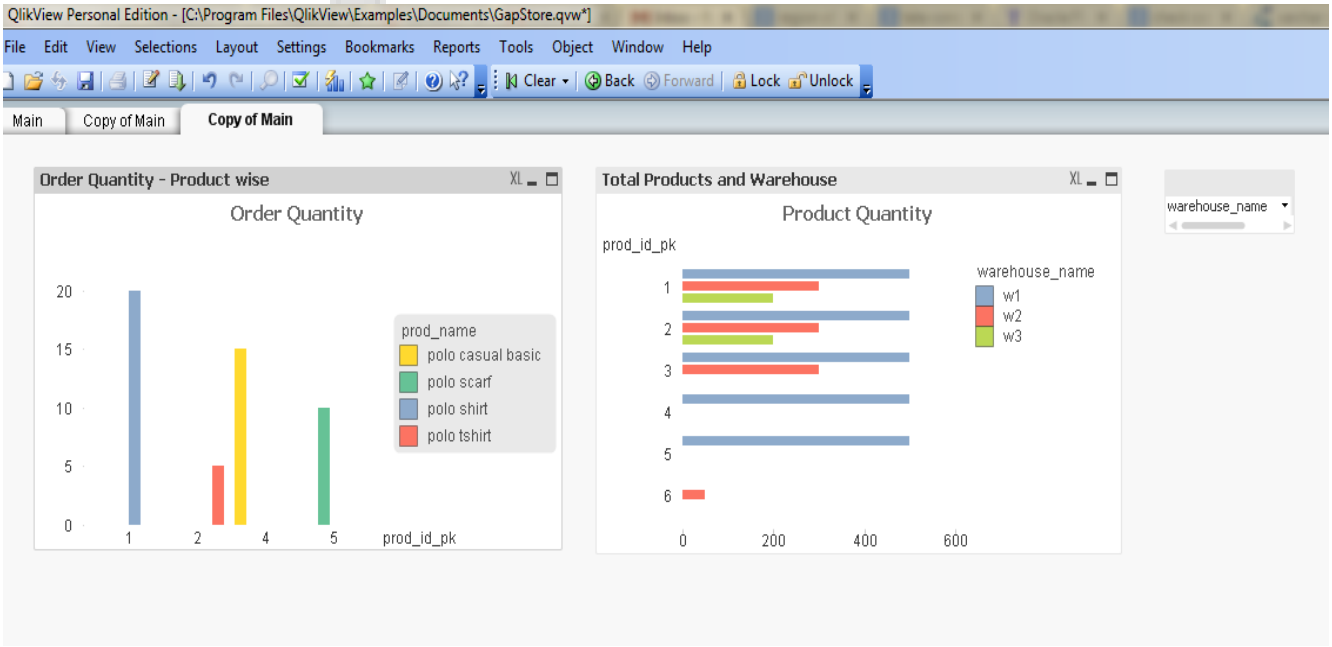
Proper privileges have been granted based upon the individual roles of Admin, Manager, Accountant, Warehouse, Returns, Shipping and Customer. Please refer to the SQL file attached for complete grant details.

11. SPECIAL DESIGN PATTERNS: BI

Business Intelligence and reporting tools are an effective way to visualize data in an efficient manner. These are a type of application software designed to retrieve, analyze, transform and report data for business intelligence. The tools generally read data that have been previously stored, often, though not necessarily, in a data warehouse or data mart.

QlikView is one of the tools used for visualizing data. With QlikView, we can run reports and create dashboards quickly to detect market changes and product sales in real time. This allows our salespeople to imediately respond to new opportunities and improve business performance.

I have used Qlikview to represent one of the views for displaying the product quantity and the warehouse that supplies these products.
Below is an example of how a bar chart would look if we try to merge the warehouse along with the product and order quantity:



Qvw file has been attached herewith for further analysis and data manipulations and visualizations.

GapStore.qvw

12. CONCLUSION

The report comprises of functional specifications and configurations, and design related prototype explained in detail. The database is designed considering data integrity and security standards. The naming conventions are given as per the standard rules and regulations. The database is designed using set business constraints and logic based upon the basic business requirements.

13. REFERENCES

Books and links References:

http://downloads.mysql.com/docs/mysql-tutorial-excerpt-5.1-en.pdf

https://dev.mysql.com/doc/refman/5.0/en/backup-and-recovery.html

http://www.tutorialspoint.com/jdbc/jdbc-stored-procedure.htm

http://www.tutorialspoint.com/sqlite/sqlite_triggers.htm

Class slides, Prof. Alex Kangoun