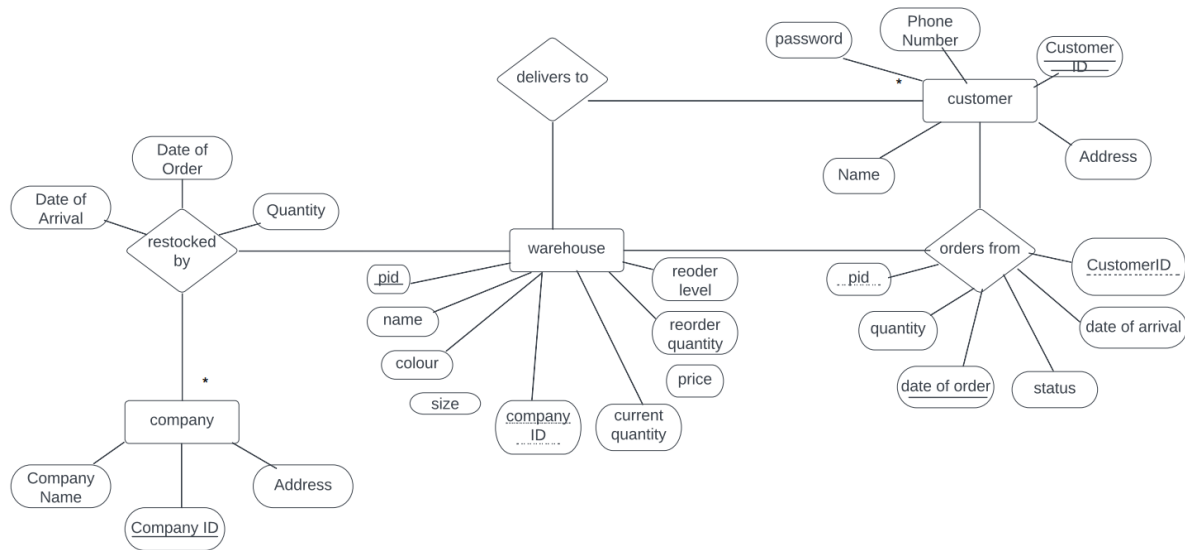


# INVENTORY MANAGEMENT SYSTEM

## ERD:



## System Requirement Specifications

### Introduction:

This project simulates a database system which keeps records of Garments supplied by all brands. This Inventory is managed centrally by a E-Commerce Company (insert name of company(admin)) and provides users all round experience of Shopping at a single destination-The (name of website).

### Problem Statement:

Creating an inventory management system that deals with Shopping transactions of Customers. The Aim is to create a legitimate source of data that can be utilized by E-Commerce websites and allow

businesses to reach more than one outlet through integrated data management.

### Goals and Scope:

As online stores are growing day by day more and more, and also the increasing the complexity of storing information of inventories and related to the inventory system, they face many related issues: availability of products, required products, etc.

This project is based on the inventory system where this application gives maximum services in a single software product that is used by the admin and the supplier. This project is based on a web app application that is sharing information on different inventories. In this project that includes Java and SQL. Java is used to design the web app for the application by which the user can interact with software applications. The SQL Server is used for creating the database in which different information will store. The main focus of this project is to give the best web app for the users and provide the many modules in a single product. Admin can view all of the information that is stored in the database through application and admin also can modify this information because the admin has full access to the system.

Database Attributes(if Primary keys, why they are primary keys):

- 1.)Unique ID ( For Unique Identification for each product)
- 2.)PID()
- 3.)Company ID(For Unique Identification for Company)
- 4.)User ID(For Unique Identification number for each customer)
- 5.)Admin ID(For Unique Identification for Each Admin)

### Hardware Requirements:

Category	Minimum	Maximum	
		Windows	MAC OS X
PROCESSOR:	Intel P-III system	Intel Core i5 and above ( Intel Core i7) OR AMD FX 4100 and above, or A6 and above	
HARD DISK:	40GB	128 Gigabytes (GB)*	128GB+Solid State Disk(SSD) with at least 20 GB of free disk space
RAM:	1GB	8GB	
PROCESSOR SPEED:	833MHz		
Wireless Networking	802.09g	802.13ax	

### Software Requirements:

Category	Minimum	Maximum
OPERATING SYSTEM:	Windows 7	Windows 10 and above

ENVIRONMENT:	Eclipse IDE	Eclipse IDE
.NET FRAMEWORK:	Version 1.0	Version 4.5.2
LANGUAGE:	JDK 1.1	JDK 1.0
BACKEND:	MYSQL 5.0, JDBC	-----
FRONTEND:	JAVA SWING	-----

### External interface Requirements: Web user interface/Web Application

#### Overview:

- It manages the information and data of the company whether small record or big detailed one to have a record of every detail in the system.
- The items are sent to different locations which need to keep records centralized in the system.
- The items are of many types and quantity therefore the system help's in keeping the items distinct and send alert time to time.
- At the time of any occasion, the demand for particular products increases at that time There should be enough quantity of items to be supplied at various places which are maintained by keeping record by the system.

Tables from ERD:

WAREHOUSE:

<u>pid</u>	name	colour	size	Company ID	Current quantity	price	Reorder quantity	Reorder level

COMPANY:

<u>Company ID</u>	Company Name	Address

CUSTOMER:

<u>CustomerID</u>	Name	Address	Password	Phone No

ORDERS:

pID	Quantity	<u>DateOfOrder</u>	status	DateOfArrival	CustomerID

RESTOCK:

Date Of Arrival	Date of Order	Quantity

### NORMALIZATION

- The database is in 1NF because no attribute of a record contains multiple entries
- It is also in 2NF because
- It is in 3NF as no primary key attribute depends on another non primary key attribute and hence is not transitively dependent on the primary key
- It is BCNF as all the attributes are only dependent on the primary key of their table

Create table commands:

Warehouse:

```
mysql> create table warehouse ( pID int NOT NULL,
```

```
-> ProductColour varchar(100),
```

```
-> ProductSize varchar(10),
```

```
-> ReorderLevel int,
```

```
-> CurrentQuantity int,
```

```
-> ProductName varchar(100),
```

- > CompanyID int,
- > PRIMARY KEY (pID),
- > FOREIGN KEY (CompanyID) REFERENCES Company(CompanyID)
- > );

Query OK, 0 rows affected (0.03 sec)

Company:

```
mysql> create table Company ( CompanyID int NOT NULL,  
    -> CompanyName varchar(100),  
    -> Address varchar(100),  
    -> PRIMARY KEY (CompanyID)  
    -> );
```

Query OK, 0 rows affected (0.01 sec)

Customer:

```
mysql> create table Customer (userID int NOT NULL,  
    -> Address varchar(100),
```

-> PhNo bigint,  
-> Name varchar(100),  
-> PRIMARY KEY (userID)  
-> );

Orders:

```
create table orders(userID int,  
  
    DateOfOrder DATE,  
  
    DateOfDelivery DATE,  
  
    Quantity int,  
  
    status varchar (30),  
  
    pID int,  
  
    CompanyID int);  
  
PRIMARY KEY (userID),  
  
PRIMARY KEY (DateOfOrder),  
  
FOREIGN KEY (pID) REFERENCES warehouse(pID),  
  
FOREIGN KEY (CompanyID) REFERENCES comapny(CompanyID)  
  
);
```

Restock:



```
mysql> create table restock(pID int,

->   Quantity int,

->   DateOfOrder DATE,

->   CompanyID int,

->   PRIMARY KEY (pID, DateOfOrder)

-> );
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> alter table restock add foreign key (CompanyID) references
company(CompanyID);
```

Query OK, 0 rows affected (0.04 sec)

Records: 0 Duplicates: 0 Warnings: 0

Queries:

1) Prices of items between 100 and 500

```
mysql> SELECT * FROM Warehouse
-> WHERE Price BETWEEN 200 AND 500;
```

pID	ProductColour	ProductSize	ReorderLevel	CurrentQuantity	ProductName	CompanyID	price	ReorderQuantity
4288	Red	XXL	10	22	TankTop	5	200	NULL
6784	White	XL	15	30	GraphicTee	4	300	NULL
9942	Black	L	10	43	BlackTop	1	400	NULL
9999	White	S	10	50	WhiteTop	2	500	NULL

```
4 rows in set (0.00 sec)
```

2) Name of customers whose names start with A

```
mysql> SELECT * FROM Customer
      -> WHERE Name LIKE 'a%';
+-----+-----+-----+-----+
| userID | Address | PhNo      | Name  |
+-----+-----+-----+-----+
|      3 | Kolkata | 9428494294 | Aryan |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

3) Order where date of order is 11/12/21

```
mysql> SELECT * FROM Orders WHERE DateOfOrder ='2021-11-12';
+-----+-----+-----+-----+-----+-----+-----+
| userID | DateOfOrder | DateOfDelivery | Quantity | status  | pID | CompanyID |
+-----+-----+-----+-----+-----+-----+-----+
|      1 | 2021-11-12 | 2021-11-15     |          1 | delivered | 2345 |          3 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

4) Order where date of delivery is 25/12/21

```
mysql> SELECT * FROM Orders WHERE DateofDelivery='2021-12-25';
+-----+-----+-----+-----+-----+-----+-----+
| userID | DateOfOrder | DateOfDelivery | Quantity | status  | pID | CompanyID |
+-----+-----+-----+-----+-----+-----+-----+
|      5 | 2021-12-20 | 2021-12-25     |          5 | delivered | 4288 |          4 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

5) Order by customer names

```
mysql> select * from customer
-> order by Name;
```

userID	Address	PhNo	Name
3	Kolkata	9428494294	Aryan
1	Pune	9424823522	Mina
5	Chennai	5829692853	Priya
2	Mumbai	8582469235	Raman
4	Delhi	8923448135	Ruby

```
5 rows in set (0.00 sec)
```

## 6) Order by product colour from warehouse

```
mysql> select * from warehouse
-> order by ProductColour;
```

pID	ProductColour	ProductSize	ReorderLevel	CurrentQuantity	ProductName	CompanyID	price	ReorderQuantity
9942	Black	L	10	43	BlackTop	1	400	NULL
2345	Lilac	M	5	29	LilacCamisole	3	100	NULL
4288	Red	XXL	10	22	TankTop	5	200	NULL
6784	White	XL	15	30	GraphicTee	4	300	NULL
9999	White	S	10	50	WhiteTop	2	500	NULL

```
5 rows in set (0.00 sec)
```

## 7) Group by

```
mysql> SELECT COUNT(userID), Address
-> FROM Customer
-> GROUP BY Address;
```

COUNT(userID)	Address
1	Kolkata
1	Pune
1	Chennai
1	Mumbai
1	Delhi

```
5 rows in set (0.00 sec)
```

## 8) Joins by cartesian product

```
mysql> SELECT warehouse.ProductName,warehouse.ProductColour,  
-> company.CompanyName,company.CompanyID  
-> FROM warehouse  
-> CROSS JOIN company;
```

ProductName	ProductColour	CompanyName	CompanyID
LilacCamisole	Lilac	PrincessPolly	5
LilacCamisole	Lilac	DollsKills	4
LilacCamisole	Lilac	Forever21	3
LilacCamisole	Lilac	Zara	2
LilacCamisole	Lilac	H&M	1
TankTop	Red	PrincessPolly	5
TankTop	Red	DollsKills	4
TankTop	Red	Forever21	3
TankTop	Red	Zara	2
TankTop	Red	H&M	1
GraphicTee	White	PrincessPolly	5
GraphicTee	White	DollsKills	4
GraphicTee	White	Forever21	3
GraphicTee	White	Zara	2
GraphicTee	White	H&M	1
BlackTop	Black	PrincessPolly	5
BlackTop	Black	DollsKills	4
BlackTop	Black	Forever21	3
BlackTop	Black	Zara	2
BlackTop	Black	H&M	1
WhiteTop	White	PrincessPolly	5
WhiteTop	White	DollsKills	4
WhiteTop	White	Forever21	3
WhiteTop	White	Zara	2
WhiteTop	White	H&M	1

25 rows in set (0.00 sec)

## 9) Inner join

```
mysql> SELECT Orders.userID, Customer.Name
-> FROM Orders
-> INNER JOIN Customer ON Orders.userID = Customer.userID;
```

userID	Name
1	Mina
5	Priya

```
2 rows in set (0.00 sec)
```

10) Like key word

```
mysql> SELECT * FROM warehouse
-> WHERE ProductName LIKE 'l%';
```

pID	ProductColour	ProductSize	ReorderLevel	CurrentQuantity	ProductName	CompanyID	price	ReorderQuantity
2345	Lilac	M	5	29	LilacCamisole	3	100	NULL

```
1 row in set (0.00 sec)
```

11) Left outer join

```
mysql> select Address, PhNo from customer
-> left outer join orders
-> on customer.userID = orders.userID;
```

Address	PhNo
Pune	9424823522
Mumbai	8582469235
Kolkata	9428494294
Delhi	8923448135
Chennai	5829692853

```
5 rows in set (0.00 sec)
```

12) Right outer join

```
mysql> select Address, PhNo from customer
->      right outer join orders
->      on customer.userID = orders.userID;
```

Address	PhNo
Pune	9424823522
Chennai	5829692853

2 rows in set (0.00 sec)

13) Subquery:

```
mysql> SELECT *
->      FROM WAREHOUSE
->      WHERE pID IN (SELECT pID
->                    FROM WAREHOUSE
->                    WHERE PRICE > 100) ;
```

pID	ProductColour	ProductSize	ReorderLevel	CurrentQuantity	ProductName	CompanyID	price	ReorderQuantity
4288	Red	XXL	10	22	TankTop	5	200	NULL
6784	White	XL	15	30	GraphicTee	4	300	NULL
9942	Black	L	10	43	BlackTop	1	400	NULL
9999	White	S	10	50	WhiteTop	2	500	NULL

14) Views:

CREATE VIEW PuneCustomers AS

SELECT Name, PhNo

FROM Customer

WHERE Address = 'Pune';

15) Views:

CREATE VIEW AboveAvg AS

SELECT ProductName, Price

FROM WAREHOUSE

WHERE Price > (SELECT AVG(Price) FROM WAREHOUSE);

16) Index:

CREATE INDEX idx\_pname

ON Customer (Name, Address);

17) Procedure:

```
create procedure customerss_orderss(IN usersid int,IN productsname varchar(20),IN  
productssize varchar(20),IN productscolor varchar(20),IN Quantitys int,IN  
Companyname varchar(20))
```

```
-> begin
```

```
-> declare productid int;
```

```
-> declare compid int;
```

```
-> declare useid int;
```

```
-> select CompanyID into Compid from Company where  
companyname=companyname;
```

```
-> select PID into productid from warehouse where productcolour=productscolor  
and productsize=productssize and productname=productsname and  
CompanyID=compid;
```

```
-> select userID into useid from customer where userID=usersid;
```

```
-> insert into  
orders(userID,pID,DateOfOrder,Quantity,status,CompanyID)values(useid,productid,n  
ow(),Quantitys,'pending',compid);
```

-> end

-> &&

Query OK, 0 rows affected (0.02 sec)

#### 18) Procedure:

```
create procedure dispatched(IN productid int,IN compid int,IN DateOrder datetime,IN  
userid int)
```

```
begin
```

```
declare Quantitys int;
```

```
select Quantity into Quantitys from orders where CompanyId=compid and pID=productid  
and DateOfOrder=DateOrder and userID= userID;
```

```
update orders set status='Dispatched' where CompanyId=compid and pID=productid  
and DateOfOrder=DateOrder and userID= userID;
```

```
update orders set DateOfDelivery=now() where CompanyId=compid and pID=productid  
and DateOfOrder=DateOrder and userID= userID;
```

```
update warehouse set CurrentQuantity=(CurrentQuantity-Quantitys) where  
CompanyId=compid and PID=productid;
```

```
end
```

```
&&
```

#### 19) Trigger

```
create trigger restockorders after insert on orders
```

```
-> for each row
```

```
-> begin
```

```
-> declare newcurrquantity int;
```



```
mysql> select * from warehouse where ProductColour = 'Lilac';
```

pID	ProductColour	ProductSize	ReorderLevel	CurrentQuantity	ProductName	CompanyID	price	ReorderQuantity
2345	Lilac	M	5	29	LilacCamisole	3	100	NULL

1 row in set (0.00 sec)

