Database Design of
Inventory Management System
AS
Lab Report on
Database Management System

Submitted by: Submitted to:

Raunak Tuladhar   Hari Saran Manandhar

BSc.CSIT 4th Semester

Section: A

Roll No. : 28569/078

I. <u>Introduction</u>

Inventory management is a critical element of the supply chain which tracks the inventory from manufacturers to warehouses and from these facilities to a point of sale. The goal of inventory management is to have the right products in the right place at the right time. Inventory management tracks how much physical inventory you have in your organization. It monitors stock at other locations, such as distributors or subcontractors. When you have clear visibility into your inventory, you know when to order, where to store it, and when you need to stop selling. Inventory management requires inventory visibility knowing when to order, how much to order and where to store stock. Multichannel order fulfillment operations typically have inventory spread across many places throughout the supply chain. Businesses need an accurate view of inventory to guarantee fulfillment of customer orders, reduce shipment turnaround times, and minimize stockouts, oversells and markdowns.

II. <u>Objectives:</u>
1. Material availability
2. Minimum Wastage
3. Maintaining Stock
4. Cost-Effective Storage
5. Minimum Cost Value for Inventories
6. Better Product Sales
7. Better Customer Services

1. <u>Material Availability:</u>

The primary objective of the inventory management system is to make sure material availability. With sufficient inventory, the production process can continue. Lack of material can cause considerable losses to the market and company profits. It will also reduce sales which will adversely affect production.

2. <u>Minimum Wastage:</u>

A successful inventory management system always helps with minimum wastage which results to minimum losses. Monitoring, controlled checks, and regulation of unmanageable wastage are some of the objectives of inventory management software that help businesses prevent and reduce waste production.

3. Maintaining Stock:

Supplies must be available for all stages of production, from raw materials to finished goods. You have to make sure that inventory consists of enough necessary material on hand to meet client demand without cutting corners.

4. Cost-Effective Storage:

Cost-effective storage is a way to reduce inventory costs. With the right management techniques, companies can avoid overstock, and duplicate stocking orders and can help to minimize storage costs.

5. Minimum Cost Value for Inventories:

An organization may negotiate discounts and other incentives for a lower overall cost when buying regular and bulk products. Additionally, it can also help to reduce inventory carrying costs.

6. <u>Better Product Sales:</u>

Inventory management can be used to calculate the volume of product sales. Sales is one of the process's most essential phases. That's why understanding the present sales condition and making future assumptions from the analysis are critical elements in a successful prediction.

7. <u>Better Customer Services:</u>

With proper inventory management, customers can gain confidence and receive timely delivery and a speedy production process. It will boost the brand image as well.

III.   <u>Entities:</u>

In Inventory Management system we have created the database as Inventory under which we have created the Entities as follows:

1. Supplier
2. Product
3. Customers
4. Orders
5. Stock

IV. Schema Diagram:

The Schema Diagram for Inventory Management System is as follows:

ims_db stock
- StockID : char(5)
- ProductID : char(5)
- StockQuantity : int(11)
- ReorderLevel : int(11)
- MinimumOrderQuantity : int(11)

ims_db customer
- CustomerID : char(5)
- CustomerName : varchar(50)
- CustomerAddress : varchar(50)
- CustomerPhoneNo : char(20)
- CustomerEmail : varchar(50)
- CustomerDOB : date

ims_db product
- ProductID : char(5)
- ProductName : varchar(50)
- Price : decimal(10,2)
- Category : char(5)
- SupplierID : char(5)

ims_db supplier
- SupplierID : char(5)
- SupplierName : varchar(50)
- SupplierAddress : varchar(50)
- SupplierPhoneNo : char(20)
- SupplierEmail : varchar(50)

ims_db orders
- OrderID : char(5)
- OrderDate : date
- CustomerID : char(5)
- ProductID : char(5)
- OrderQuantity : decimal(10,2)

V. SQL Queries

a) Create Database:

First we create a database named 'ims_db'.

```
1 CREATE DATABASE ims_db
```

b) Create Table:

Now we create the entities present in 'ims_db'. The entities are created as follows:

```
1 CREATE TABLE Supplier(
2     SupplierID char(5) PRIMARY KEY NOT NULL,
3     SupplierName varchar(50) NOT NULL,
4     SupplierAddress varchar(50) NOT NULL,
5     SupplierPhoneNo char(20) NOT NULL,
6     SupplierEmail varchar(50) NOT NULL
7     );
```

```
1 CREATE TABLE Product(
2     ProductID char(5) PRIMARY KEY NOT NULL,
3     ProductName varchar(50) NOT NULL,
4     Price decimal(10, 2) NOT NULL,
5     Category char(5) NOT NULL,
6     SupplierID char(5) NOT NULL,
7     FOREIGN KEY (SupplierID) REFERENCES supplier (SupplierID)
8     );
```

```
1  CREATE TABLE Customer(
2      CustomerID char(5) NOT NULL PRIMARY KEY,
3      CustomerName varchar(50) NOT NULL,
4      CustomerAddress varchar(50) NOT NULL,
5      CustomerPhoneNo char(20) NOT NULL,
6      CustomerEmail varchar(50) NOT NULL,
7      CustomerDOB date
8      );
```

```
1  CREATE TABLE Orders(
2      OrderID char(5) NOT NULL,
3      OrderDate date NOT NULL,
4      CustomerID char(5) NOT NULL,
5      ProductID char(5) NOT NULL,
6      OrderQuantity decimal(10, 2) NOT NULL,
7      FOREIGN KEY(CustomerID) REFERENCES customer (CustomerID),
8      FOREIGN KEY(ProductID) REFERENCES product (ProductID)
9      );
10
```

```
1  CREATE TABLE Stock(
2      StockID char(5) PRIMARY KEY NOT NULL,
3      ProductID char(5) NOT NULL,
4      StockQuantity int,
5      ReorderLevel int,
6      MinimumOrderQuantity int,
7      FOREIGN KEY (ProductID) REFERENCES product (ProductID)
8      );
```

The structure of database 'ims_db' after creating the entities are as follows:

Table	Action						Rows	Type	Collation	Size	Overhead
customer	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
orders	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
product	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
stock	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
supplier	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
5 tables	Sum							0	InnoDB	utf8mb4_general_ci	128.0 KiB	0 B

The structure of each entities are as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	SupplierID 🔑	char(5)	utf8mb4_general_ci		No	None			Change	Drop	More
2	SupplierName	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More
3	SupplierAddress	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More
4	SupplierPhoneNo	char(20)	utf8mb4_general_ci		No	None			Change	Drop	More
5	SupplierEmail	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More

| #	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	ProductID 🔑	char(5)	utf8mb4_general_ci		No	None			Change	Drop	More
2	ProductName	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More
3	Price	decimal(10,2)			No	None			Change	Drop	More
4	Category	char(5)	utf8mb4_general_ci		No	None			Change	Drop	More
5	SupplierID 🔑	char(5)	utf8mb4_general_ci		No	None			Change	Drop	More

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	CustomerID 🔑	char(5)	utf8mb4_general_ci		No	None			Change	Drop	More
2	CustomerName	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More
3	CustomerAddress	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More
4	CustomerPhoneNo	char(20)	utf8mb4_general_ci		No	None			Change	Drop	More
5	CustomerEmail	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More
6	CustomerDOB	date			Yes	NULL			Change	Drop	More

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	OrderID	char(5)	utf8mb4_general_ci		No	None			Change	Drop	More
2	OrderDate	date			No	None			Change	Drop	More
3	CustomerID 🔑	char(5)	utf8mb4_general_ci		No	None			Change	Drop	More
4	ProductID 🔑	char(5)	utf8mb4_general_ci		No	None			Change	Drop	More
5	OrderQuantity	decimal(10,2)			No	None			Change	Drop	More

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 StockID 🔑	char(5)	utf8mb4_general_ci		No	None			✏ Change ⊖ Drop More
☐	2 ProductID 🔑	char(5)	utf8mb4_general_ci		No	None			✏ Change ⊖ Drop More
☐	3 StockQuantity	int(11)			Yes	NULL			✏ Change ⊖ Drop More
☐	4 ReorderLevel	int(11)			Yes	NULL			✏ Change ⊖ Drop More
☐	5 MinimumOrderQuantity	int(11)			Yes	NULL			✏ Change ⊖ Drop More

c) **Alter Tables:**

Here, in the Orders entity, OrderID is the primary key. Since we have not denoted it as the Primary Key, so we are going to alter it by the following SQL Query:

```
1  ALTER TABLE orders
2  ADD CONSTRAINT PRIMARY KEY (OrderID)
```

So the result will be:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 OrderID 🔑	char(5)	utf8mb4_general_ci		No	None			✏ Change ⊖ Drop More
☐	2 OrderDate	date			No	None			✏ Change ⊖ Drop More
☐	3 CustomerID 🔑	char(5)	utf8mb4_general_ci		No	None			✏ Change ⊖ Drop More
☐	4 ProductID 🔑	char(5)	utf8mb4_general_ci		No	None			✏ Change ⊖ Drop More
☐	5 OrderQuantity	decimal(10,2)			No	None			✏ Change ⊖ Drop More

d) Insert Data:

Now we insert the values into the entities:

```sql
1  INSERT INTO supplier
2  VALUES('S1', 'HP World', 'Sector 11 Dwarka', '9999001001', 'hpworld@gmail.com'),
3  ('S2', 'Medcare', 'SC Kanpur, NP Road', '8887001222', 'medcare@gmail.com'),
4  ('S3', 'Retail Shoppers', 'B1/45, Janakpuri', '3337001333', 'retailshop@yahoo.com.in'),
5  ('S4', 'IT Pickup', 'DLF, Phase 5', '703331343', 'itpickup@gmail.com'),
6  ('S5', 'Home Finisher', 'C2-431, Supertech', '99003388', 'homefinish@outlook.com')
```

```sql
1  INSERT INTO product
2  VALUES('P1', 'HP Laptop and Charger', 6000, 'IT', 'S1'),
3  ('P2', 'Mixer Grinder', 10000, 'HA', 'S5'),
4  ('P3', 'Mobile Phone', 4000, 'IT', 'S4'),
5  ('P4', 'Steamer', 400, 'HC', 'S2'),
6  ('P5', 'Chairs and Tables', 10500, 'HA', 'S3')
```

```sql
1  INSERT INTO customer
2  VALUES('C1', 'Abhishek Sharma', '7TH Cross, Janakpuri', '9088801001', 'asharma@gmail.com', '10-01-1996'),
3  ('C2', 'Rinku Sharma', 'HNO-8205', '809881077', 'rinku@gmail.com', '10-04-1999'),
4  ('C3', 'Teena Mathew', 'DLF, Phase 5', '9880601077', 'teena@yahoo.com', '29-06-1990'),
5  ('C4', 'Roshan Singh', 'Flat No. 6A, Rajdur Road', '6767601123', 'roshan@outlook.com', '20-08-1996'),
6  ('C5', 'Monika Babi', 'Pocket 2, Sector 11', '5577640107', 'monika@gmail.com', '20-07-1997')
```

```sql
1  INSERT INTO orders
2  VALUES('O1', '2018-07-15', 'C5', 'P1', 1),
3  ('O2', '2017-08-02', 'C2', 'P4', 4),
4  ('O3', '2016-01-07', 'C1', 'P2', 2),
5  ('O4', '1998-07-05', 'C4', 'P5', 7),
6  ('O5', '2018-07-15', 'C3', 'P3', 3)
```

```
1  INSERT INTO stock
2  VALUES('ST1', 'P5', 3, 4, 8),
3  ('ST2', 'P2', 2, 5, 20),
4  ('ST3', 'P4', 5, 3, 15),
5  ('ST4', 'P1', 0, 1, 10),
6  ('ST5', 'P5', 6, 2, 12)
```

After inserting the values, the entities will look like as follows:

		SupplierID	SupplierName	SupplierAddress	SupplierPhoneNo	SupplierEmail
☐	Edit Copy Delete	S1	HP World	Sector 11 Dwarka	9999001001	hpworld@gmail.com
☐	Edit Copy Delete	S2	Medcare	SC Kanpur, NP Road	8887001222	medcare@gmail.com
☐	Edit Copy Delete	S3	Retail Shoppers	B1/45, Janakpuri	3337001333	retailshop@yahoo.com.in
☐	Edit Copy Delete	S4	IT Pickup	DLF, Phase 5	703331343	itpickup@gmail.com
☐	Edit Copy Delete	S5	Home Finisher	C2-431, Supertech	99003388	homefinish@outlook.com

		ProductID	ProductName	Price	Category	SupplierID
☐	Edit Copy Delete	P1	HP Laptop and Charger	6000.00	IT	S1
☐	Edit Copy Delete	P2	Mixer Grinder	10000.00	HA	S5
☐	Edit Copy Delete	P3	Mobile Phone	4000.00	IT	S4
☐	Edit Copy Delete	P4	Steamer	400.00	HC	S2
☐	Edit Copy Delete	P5	Chairs and Tables	10500.00	HA	S3

	CustomerID	CustomerName	CustomerAddress	CustomerPhoneNo	CustomerEmail	CustomerDOB
☐ 🖉 Edit ⯐ Copy ⊖ Delete	C1	Abhishek Sharma	7TH Cross, Janakpuri	9088801001	asharma@gmail.com	0000-00-00
☐ 🖉 Edit ⯐ Copy ⊖ Delete	C2	Rinku Sharma	HNO-8205	809881077	rinku@gmail.com	0000-00-00
☐ 🖉 Edit ⯐ Copy ⊖ Delete	C3	Teena Mathew	DLF, Phase 5	9880601077	teena@yahoo.com	0000-00-00
☐ 🖉 Edit ⯐ Copy ⊖ Delete	C4	Roshan Singh	Flat No. 6A, Rajdur Road	6767601123	roshan@outlook.com	0000-00-00
☐ 🖉 Edit ⯐ Copy ⊖ Delete	C5	Monika Babi	Pocket 2, Sector 11	5577640107	monika@gmail.com	0000-00-00

	StockID	ProductID	StockQuantity	ReorderLevel	MinimumOrderQuantity
☐ 🖉 Edit ⯐ Copy ⊖ Delete	ST1	P5	3	4	8
☐ 🖉 Edit ⯐ Copy ⊖ Delete	ST2	P2	2	5	20
☐ 🖉 Edit ⯐ Copy ⊖ Delete	ST3	P4	5	3	15
☐ 🖉 Edit ⯐ Copy ⊖ Delete	ST4	P1	0	1	10
☐ 🖉 Edit ⯐ Copy ⊖ Delete	ST5	P5	6	2	12

	OrderID	OrderDate	CustomerID	ProductID	OrderQuantity
☐ 🖉 Edit ⯐ Copy ⊖ Delete	O1	2018-07-15	C5	P1	1.00
☐ 🖉 Edit ⯐ Copy ⊖ Delete	O2	2017-08-02	C2	P4	4.00
☐ 🖉 Edit ⯐ Copy ⊖ Delete	O3	2016-01-07	C1	P2	2.00
☐ 🖉 Edit ⯐ Copy ⊖ Delete	O4	1998-07-05	C4	P5	7.00
☐ 🖉 Edit ⯐ Copy ⊖ Delete	O5	2018-07-15	C3	P3	3.00

e) Update Data:

As we see in the above SQL Query, while inserting the value in CustomerDOB attribute in Customer entity, the format of the date is completely wrong which results to the following:

```
1 INSERT INTO customer
2 VALUES('C1', 'Abhishek Sharma', '7TH Cross, Janakpuri', '9088801001', 'asharma@gmail.com', '10-01-1996'),
3 ('C2', 'Rinku Sharma', 'HNO-8205', '809881077', 'rinku@gmail.com', '10-04-1999'),
4 ('C3', 'Teena Mathew', 'DLF, Phase 5', '9880601077', 'teena@yahoo.com', '29-06-1990'),
5 ('C4', 'Roshan Singh', 'Flat No. 6A, Rajdur Road', '6767601123', 'roshan@outlook.com', '20-08-1996'),
6 ('C5', 'Monika Babi', 'Pocket 2, Sector 11', '5577640107', 'monika@gmail.com', '20-07-1997')
```

←T→			CustomerID	CustomerName	CustomerAddress	CustomerPhoneNo	CustomerEmail	CustomerDOB
☐	🖉 Edit	📋 Copy ⊖ Delete	C1	Abhishek Sharma	7TH Cross, Janakpuri	9088801001	asharma@gmail.com	0000-00-00
☐	🖉 Edit	📋 Copy ⊖ Delete	C2	Rinku Sharma	HNO-8205	809881077	rinku@gmail.com	0000-00-00
☐	🖉 Edit	📋 Copy ⊖ Delete	C3	Teena Mathew	DLF, Phase 5	9880601077	teena@yahoo.com	0000-00-00
☐	🖉 Edit	📋 Copy ⊖ Delete	C4	Roshan Singh	Flat No. 6A, Rajdur Road	6767601123	roshan@outlook.com	0000-00-00
☐	🖉 Edit	📋 Copy ⊖ Delete	C5	Monika Babi	Pocket 2, Sector 11	5577640107	monika@gmail.com	0000-00-00

So to update the its value, we use 'UPDATE' to modify the values.

```
1 UPDATE customer
2 SET CustomerDOB = '1996-01-10'
3 WHERE CustomerID = 'C1'
```

```
1 UPDATE customer
2 SET CustomerDOB = '1999-04-10'
3 WHERE CustomerID = 'C2'
```

```
1 UPDATE customer
2 SET CustomerDOB = '1990-06-29'
3 WHERE CustomerID = 'C3'
```

```
1 UPDATE customer
2 SET CustomerDOB = '1996-08-20'
3 WHERE CustomerID = 'C4'
```

```
1 UPDATE customer
2 SET CustomerDOB = '1997-07-20'
3 WHERE CustomerID = 'C5'
```

The result is:

				CustomerID	CustomerName	CustomerAddress	CustomerPhoneNo	CustomerEmail	CustomerDOB
☐	✏ Edit	⯎ Copy	⊖ Delete	C1	Abhishek Sharma	7TH Cross, Janakpuri	9088801001	asharma@gmail.com	1996-01-10
☐	✏ Edit	⯎ Copy	⊖ Delete	C2	Rinku Sharma	HNO-8205	809881077	rinku@gmail.com	1999-04-10
☐	✏ Edit	⯎ Copy	⊖ Delete	C3	Teena Mathew	DLF, Phase 5	9880601077	teena@yahoo.com	1990-06-29
☐	✏ Edit	⯎ Copy	⊖ Delete	C4	Roshan Singh	Flat No. 6A, Rajdur Road	6767601123	roshan@outlook.com	1996-08-20
☐	✏ Edit	⯎ Copy	⊖ Delete	C5	Monika Babi	Pocket 2, Sector 11	5577640107	monika@gmail.com	1997-07-20

f) Create View:

Now, we create view

```
1  CREATE VIEW view1 AS
2  SELECT product.ProductID, product.SupplierID
3  FROM product
4  WHERE Category = 'IT'
```

```
1  CREATE VIEW view2 AS
2  SELECT orders.CustomerID, orders.ProductID
3  FROM orders
4  WHERE OrderID = 'O1'
```

The result will be as follows:
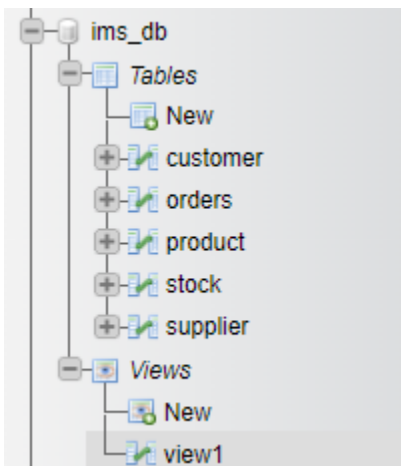
				ProductID	SupplierID
☐	Edit	Copy	Delete	P1	S1
☐	Edit	Copy	Delete	P3	S4

ims_db
- Tables
  - New
  - customer
  - orders
  - product
  - stock
  - supplier
- Views
  - New
  - view1
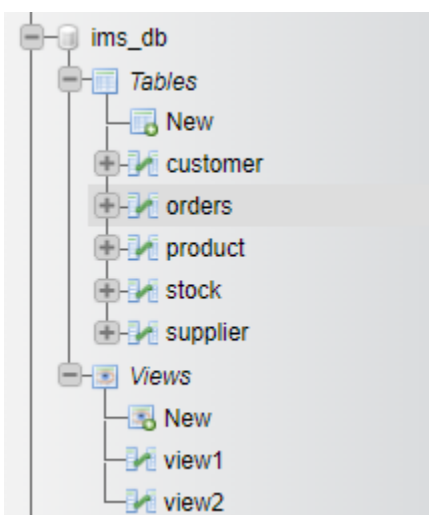
				CustomerID	ProductID
☐	Edit	Copy	Delete	C5	P1

ims_db
- Tables
  - New
  - customer
  - orders
  - product
  - stock
  - supplier
- Views
  - New
  - view1
  - view2

g) Drop View:

If we want to delete a view then we can use the SQL Query as follows:

```
1  DROP VIEW view1
```

which results to:
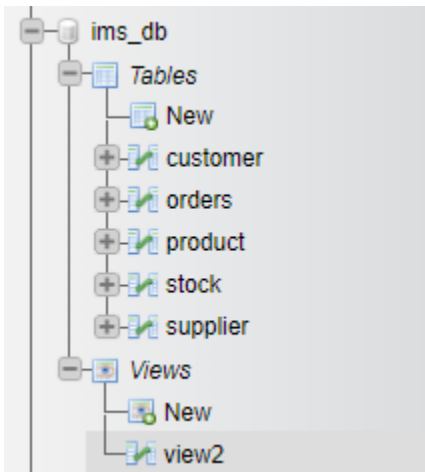
h) **Join:**

To join two tables we use the following SQL Query:

```
1 SELECT product.ProductID, product.ProductName, supplier.SupplierID
2 FROM product
3 INNER JOIN supplier ON product.SupplierID = supplier.SupplierID
```

We get the result as follows:

ProductID	ProductName	SupplierID
P1	HP Laptop and Charger	S1
P2	Mixer Grinder	S5
P3	Mobile Phone	S4
P4	Steamer	S2
P5	Chairs and Tables	S3

Similarly:

```
1 SELECT OrderID, OrderDate, CustomerName, CustomerAddress, CustomerPhoneNo, ProductName, Price,
  OrderQuantity, Price*OrderQuantity AS 'AMT'
2 FROM orders
3 INNER JOIN customer
4 ON orders.CustomerID = customer.CustomerID
5 INNER JOIN product
6 ON orders.ProductID = product.ProductID
7 INNER JOIN supplier
8 ON supplier.SupplierID = product.SupplierID
9 ORDER BY OrderDate
```

The result will be:

OrderID	OrderDate ▲ 1	CustomerName	CustomerAddress	CustomerPhoneNo	ProductName	Price	OrderQuantity	AMT
O4	1998-07-05	Roshan Singh	Flat No. 6A, Rajdur Road	6767601123	Chairs and Tables	10500.00	7.00	73500.0000
O3	2016-01-07	Abhishek Sharma	7TH Cross, Janakpuri	9088801001	Mixer Grinder	10000.00	2.00	20000.0000
O2	2017-08-02	Rinku Sharma	HNO-8205	809881077	Steamer	400.00	4.00	1600.0000
O1	2018-07-15	Monika Babi	Pocket 2, Sector 11	5577640107	HP Laptop and Charger	6000.00	1.00	6000.0000
O5	2018-07-15	Teena Mathew	DLF, Phase 5	9880601077	Mobile Phone	4000.00	3.00	12000.0000

Content: