EXPORT COMPANY

Group no: 17

Name:
Jignasuben R. Vekariya (002981797)
Nupur Bhavesh Shah (001077247)

PROBLEM DEFINITION

- The primary objective of this study was to design and implement a relational database that is industry ready for application for Export companies for use by processing data from Agents , Manufactures and Distributors including goods and payments of the same.

- There is huge amounts of data generation, and data can be reused by the properties using a relational database.

- This relational database reduces data input process time by 75% and result in better processing of goods over the world benefits across the industry.

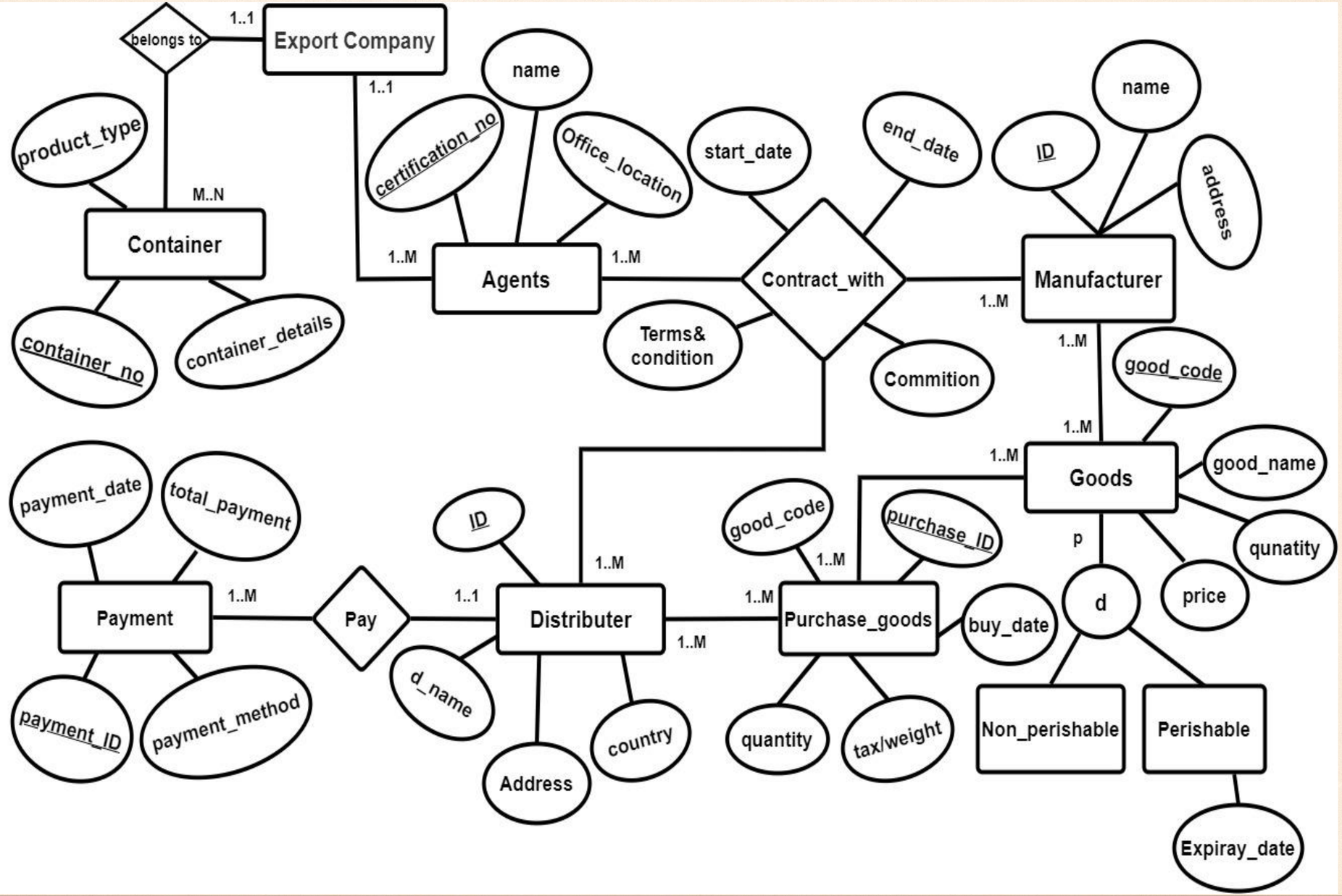
- This information also allows to track the export data and keep track of the revenue that is generated by the same.

- The database was inputed taking the data fields of a sector and agents and manufacturer and the products they distribute through the distributors.

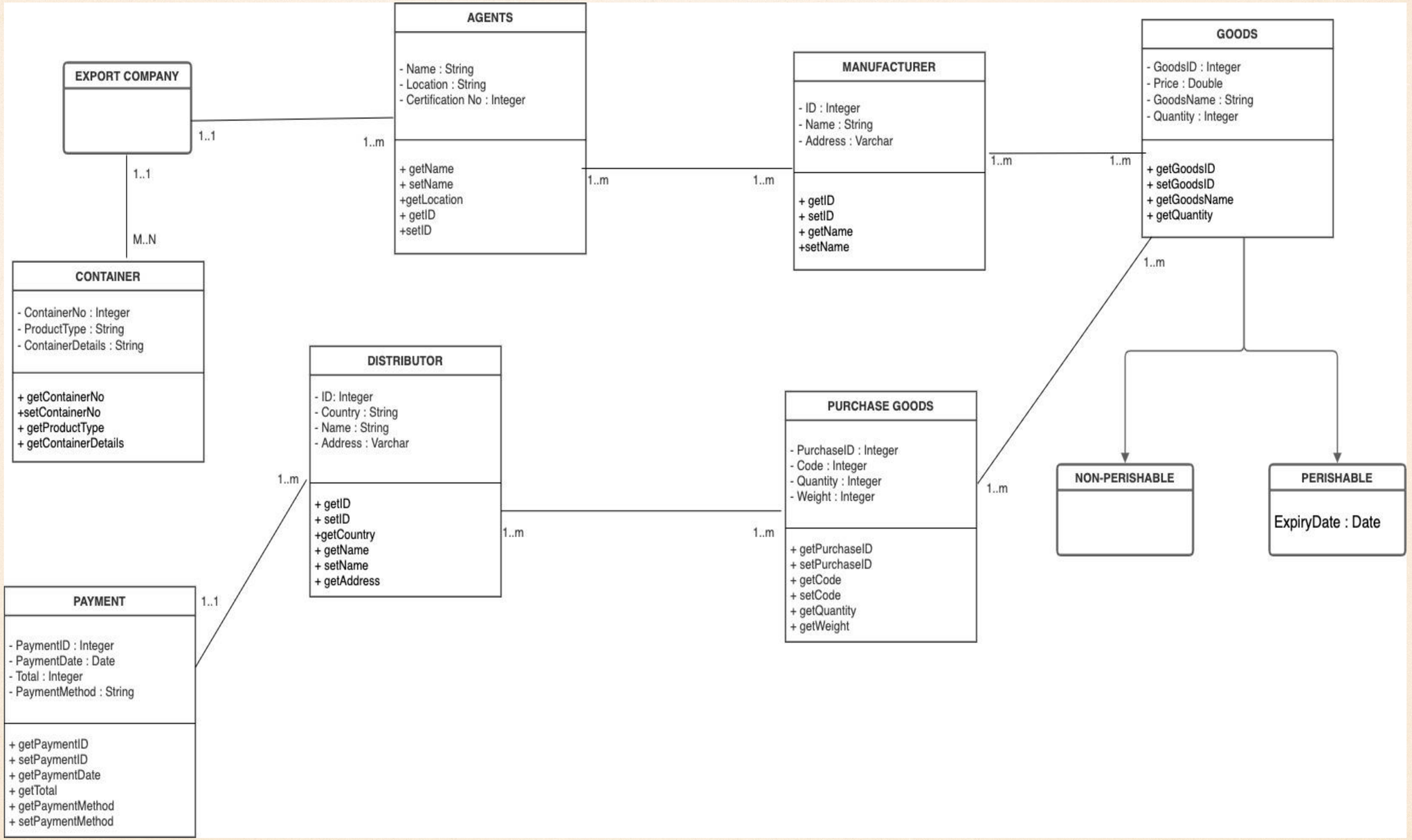
REQUIREMENTS

- The company has its own many containers for shipment
- All container owns by only one company.
- Each agent is responsible for contracts with multiple manufactures and distributers. Agent should make sure the conditions and time-period of the contract and commission will be given for the same.
- The manufacturer can supply one to many numbers and type of goods for export purpose.
- The distributor can import one to many for his/her country from same or different manufacturer.
- The distributer can make one to many payments which are under on his/her ID.
- Each payment can be made by only that one distributer.

EER DIAGRAM


belongs to
1..1
Export Company
1..1
name
certification_no
Office_location
product_type
M..N
Container
container_no
container_details
1..M
Agents
1..M
start_date
end_date
Contract_with
Terms& condition
Commition
1..M
Manufacturer
name
ID
address
1..M
good_code
1..M
1..M
Goods
good_name
qunatity
price
p
d
buy_date
Non_perishable
Perishable
Expiray_date
good_code
purchase_ID
1..M
1..M
Purchase_goods
quantity
tax/weight
1..M
1..M
Distributer
ID
d_name
Address
country
1..1
Pay
1..M
Payment
payment_date
total_payment
payment_ID
payment_method

UML DESIGN


EXPORT COMPANY
AGENTS
- Name : String
- Location : String
- Certification No : Integer
+ getName
+ setName
+getLocation
+ getID
+setID
MANUFACTURER
- ID : Integer
- Name : String
- Address : Varchar
+ getID
+ setID
+ getName
+setName
GOODS
- GoodsID : Integer
- Price : Double
- GoodsName : String
- Quantity : Integer
+ getGoodsID
+ setGoodsID
+ getGoodsName
+ getQuantity
CONTAINER
- ContainerNo : Integer
- ProductType : String
- ContainerDetails : String
+ getContainerNo
+setContainerNo
+ getProductType
+ getContainerDetails
DISTRIBUTOR
- ID: Integer
- Country : String
- Name : String
- Address : Varchar
+ getID
+ setID
+getCountry
+ getName
+ setName
+ getAddress
PURCHASE GOODS
- PurchaseID : Integer
- Code : Integer
- Quantity : Integer
- Weight : Integer
+ getPurchaseID
+ setPurchaseID
+ getCode
+ setCode
+ getQuantity
+ getWeight
NON-PERISHABLE
PERISHABLE
ExpiryDate : Date
PAYMENT
- PaymentID : Integer
- PaymentDate : Date
- Total : Integer
- PaymentMethod : String
+ getPaymentID
+ setPaymentID
+ getPaymentDate
+ getTotal
+ getPaymentMethod
+ setPaymentMethod
1..1
1..m
M..N

Mapping Conceptual to Relation Model:

Agents (certification_no, agent_name, office_location)
- Certification_no is primary key for agent table.
- Constraint: NOT NULL applied on certification_no

Manufacturer (ID, name, address)
- ID is primary key for manufacturer table.
- Constraint: NOT NULL applied on ID

Contract (certification_no, ID, start_date, end_date, commition, terms&condition)
- Certification_no is primary key. ID is foreign key form Manufacturer(ID) and Distributer(ID).
- Constraints: NOT NULL is applied on certification_no. ON UPDATE CASECADE is applied on ID.

Goods (good_code, good_name, quantity, price)
- Good_code is primary key
- Constraints: Not null applied on good_code.

Supply_from (good_code, manufacturer_ID)
- Good_code is foreign key from the Goods(good_code) table and manufacturer_ID is foreign key from the Manufacturer(ID) table.
- Constraints: ON UPDATE/ DELETE CASECADE is applied on both field

Perishable (perishable_code, expiray_date)
- perishable_code is super key from Goods(good_code) table

Unperishable (unperishable_code)
- unperishable _code is super key from Goods(good_code) table

Distributer (ID, d_name, address, country)
- ID is primary key for distributer table.
- Constraint: NOT NULL applied on ID

Purchase_goods (purchase_ID, good_code, buy_date, quantity, tax/weight, distributer_ID)
- Purchase_ID is primary key, good_code is foreign key from Goods table and distributer_ID is foreign key from the Distributer(ID) table.
- Constraint: NOT NULL is applied on all the keys and tax/weight. ON UPDATE CASECADE is applied on distributer_ID.

Payment (payment_ID, payment_date, total_payment, payment_method, distributer_ID)
- Payment_ID is primary key and distributer_ID is foreign key from the distributer table.
- Constraints: NOT NULL applied on both keys and total_payment. ON UPDATE CASECADE applied on distributer_ID.

Container (container_no, container_details, product_type)
- Container_no is primary key.
- Constraint: NOT NULL applied on container_no and product_type.

Implementation of Relation Model via MySQL and NoSQL

Query 1 : Retrieve the details of agents who has office location in Massachusetts

```
use tradedata
select * from agents
where office_location like ‘%MA';
```

Query 2 : Calculate the Number of Perishable Goods and Nonperishable Goods

```
select good_type,
count(*) as amount
from goods
group by good_type;
```

Query 3 : Retrieve the purchase details, count total amount without tax, agents and goods names from the database

```
select
    p.purchase_id,
    g.good_name,
    p.quantity_in_tons,
    g.price_in$_perTons*p.quantity_in_tons AS Total_Price,
    p.buy_date,
    d.dis_name,
    a.agent_name
from purchase_good p
join agents a
    ON p.agent_id=a.certification_no
join goods g
    ON p.good_code=g.good_code
join distributer d
    ON p.dis_id=d.Id;
Select Goodscode, Dis id, Agent id, Quantity, tax
From purchase goods
Where tax > 2
Order by tax desc
Limit 7
```

Query 4 : Retrieve the top 10 distributers and their agents names with purchased goods having a tax higher than 5%

```
Select d.dis_name, a.agent_name, g.good_name,  p.tax_perTons AS Tax
from purchase_good p
join distributer d
    ON p.dis_id=d.Id
join agents a
    ON p.agent_id=a.certification_no
join goods g
    ON p.good_code=g.good_code
Where p.tax_perTons >5
Order by p.tax_perTons desc
Limit 10;
```

Query 5 : Company want to give reminder those manufacturer whose contract with their agent will be end with in 15 months

So, they wants to view all the details of manufacturer and their agents as well as contract details

```
create view reminder AS
select a.agent_name,m.man_name,
        c.start_date,
       c.end_date,ceiling(datediff(c.end_date, curdate())/30) as duration_in_months,
       c.commition_in_per
from contract c
join agents a ON a.certification_no=c.certification_no
join manufacturer m ON m.Id=c.man_id
where datediff(c.end_date, curdate()) < 460 ;
```

Query 6 : Company wants to send notice to those agents who take more than 25% commission from their manufacturer

```
create view notice as
select * from agents
where certification_no In (
                    select distinct certification_no
                from contract
                where commmsion_in_per>25);
```

Query 7 : Management wants to check the stock of goods and status. make a procedure for this question

```
 DELIMITER $$
CREATE PROCEDURE GetGoodStock(
)
BEGIN
    create temporary table stocks
    select quantity_in_tons as quantity, IF(quantity_in_tons<50,'Out of stock','Stock Available') AS Stock
    from goods;
    SELECT * FROM stocks;
END$$
DELIMITER ;
CALL GetGoodStock()
```

NoSQL Implementation:

Database Access via MongoDB :

Tables are imported and queries are created in MongoDB. The following queries were done:

Query 1 : Details of the distributors who are from London, Uk

	_id ObjectId	ID String	Name String	Address String	Country String
1	61b64f2558709f3419deb7d1	"998"	"Asa "	"London"	"UK"
2	61b64f2558709f3419deb800	"345"	"Edena"	"London"	"UK"
3	61b64f2558709f3419deb805	"43"	"Jordana"	"London"	"UK"

Query 2 : Retrieve the goods details with a price greater than 50000

```
[{$match: {
 Price: {$gt: 50000}}},
{$project: { _id: 0,'Goods name': 1,'Quantity ': 1,Price: 1}}]
```

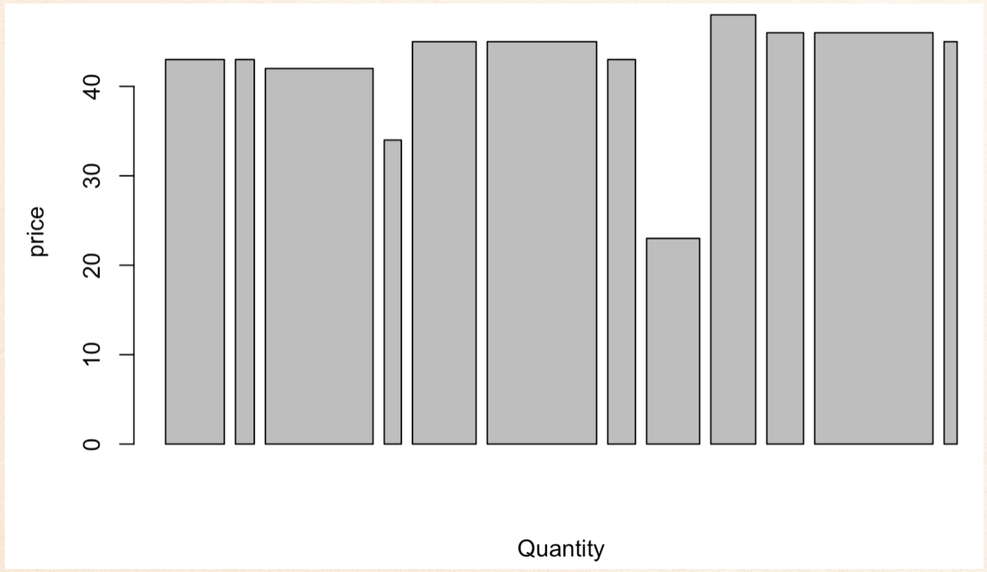
Query 3 : Retrieve the average quantity and price according to their goods division

```
[{  $group: {_id: '$Goods division',
        avg_goods: {
           $avg: '$Quantity '   },
        avg_price: {   $avg: '$Price'} }}]
```

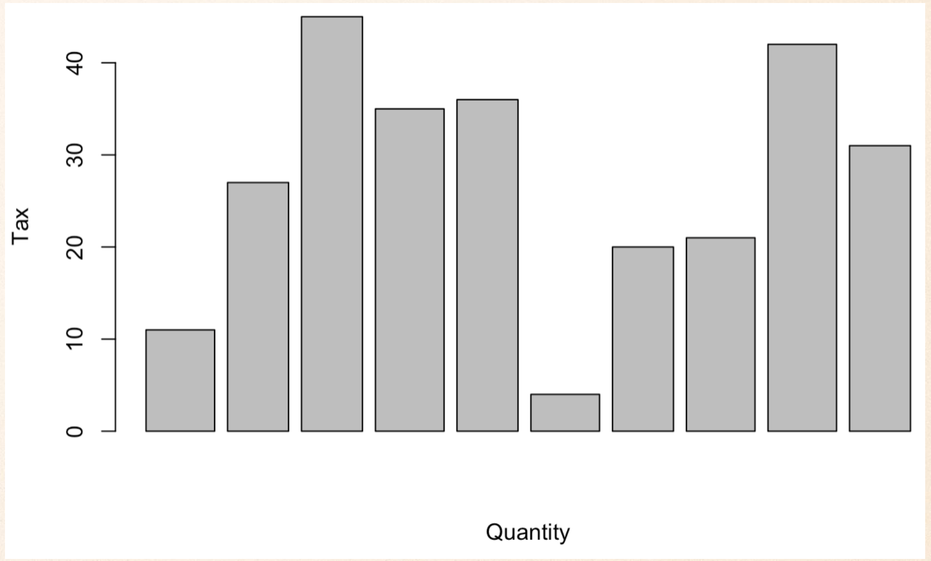
Database Access via R Programing

The database is accessed using R programming and visualization of analyzed data is shown below.
The connection of MySQL to R is done using ODBC, followed by connection to run and fetch all from query to plot the bar graph for the analytics

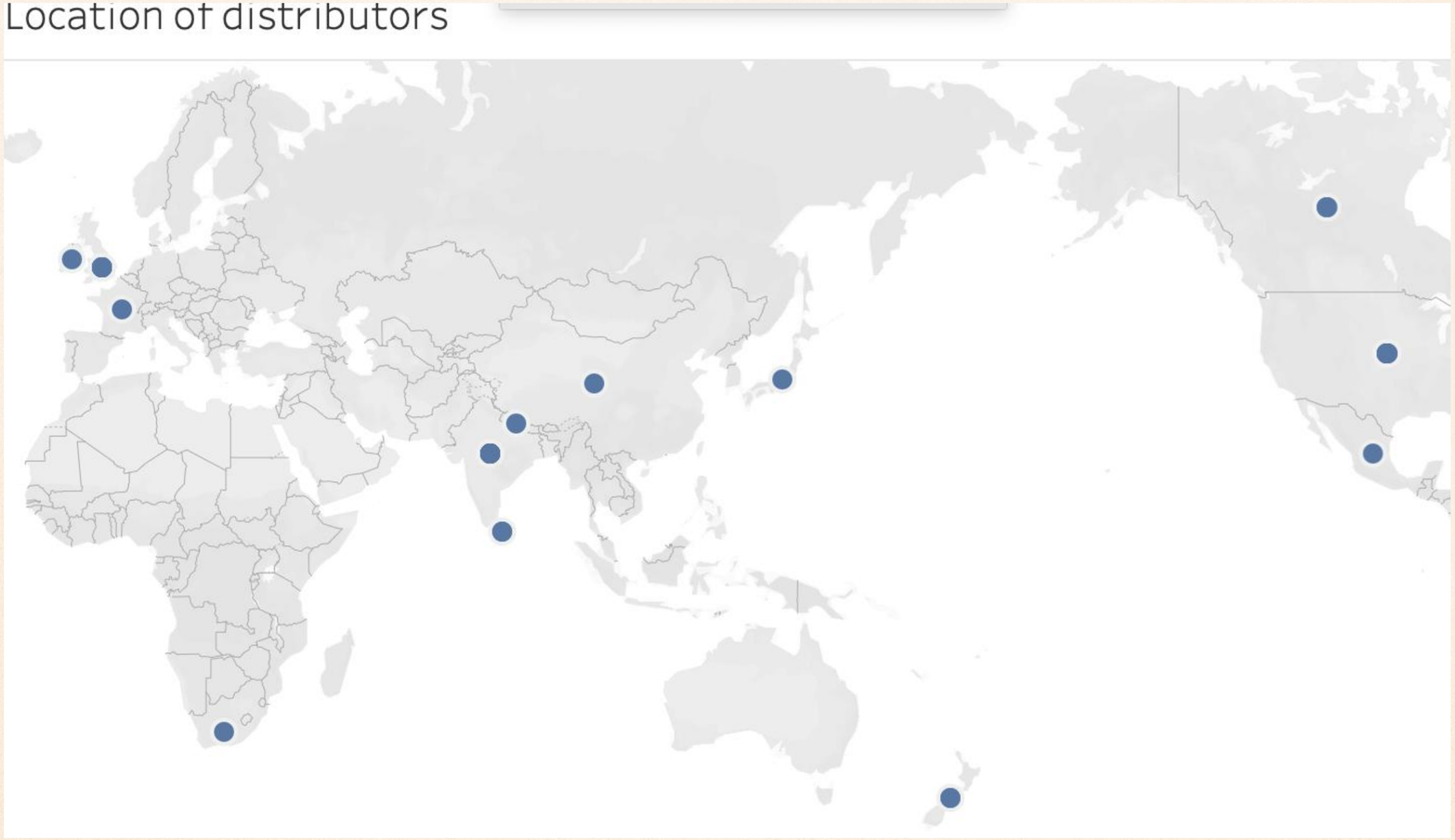
Graph 1 : Change of price with quantity



Graph 2 : Export amount of products tax and quantity



The database was also linked to Tableau for geo-mapping of location of distributors by running custom SQL Query on Database via Tableau



Summary and Recommendation

The Export database designed on MySQL Workbench is an industry with huge data that can be defined with all the cites over the world and the database implementation is done for the data of the products, agents, manufacturers and containers. This implementation of the data will help with organizing the data the is processed to be analyzed and makes the structure easier to handle. A part of this database is also shown in the report using R programming to show a defined graph of the details involved.

In this dataset the improvement that we can add would be define trends for the export process accordingly and place orders automatically when required by the agents and define the warehouse products for the manufactures.