

Final Project Report

Abdul Moiz Asif - 14090

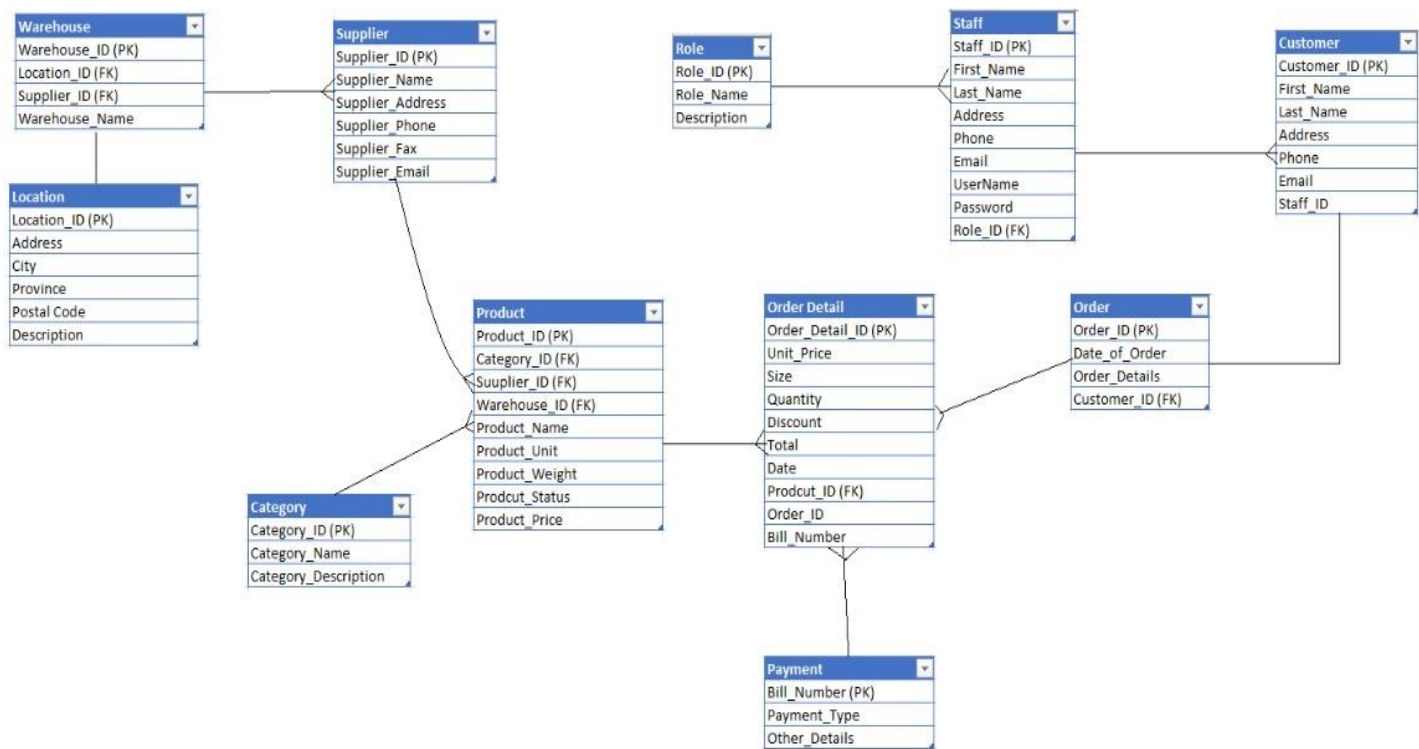
Hassan Shah Lakhani – 14856

This was our project which took the place of the final exam. The topic we were assigned was Inventory Management (To Optimize Sales and Manufacturing). Inventory Management is the approach to sourcing, storing, selling both raw material and finished goods. This is the most important part in the supply chain where you have to maintain the right amount of stock, the right cost, the right price at the right time and place.

Although It was a very general topic, we took it upon ourselves to make the most of this topic. First of all, we assumed that the Inventory Management System to be of a store which gets its supplies from the suppliers. The components we selected were 1, 2, 5 and 6 (Track 2) because we felt these were areas we wanted to work on.

Moving on to Component 1, where we had to acquire Business Knowledge for the domain we were going to work on. We took the help of YouTube videos, Wikipedia articles and Blogs. We further looked at different social media entities which provided information related to Inventory Management like Prima Solutions and Hub Works App. Aside from that, we looked at different problems stakeholders faced in the Inventory Management. Through this, we were able to gather expertise in the area.

After carefully evaluating all the resources we had saved and read, we decided to come up with our own ERD which was specifically made for the optimization of problems related to the sale and manufacturing. This was the ERD which we made.



There were a total 11 tables, namely, Warehouse, Location, Supplier, Product, Category, Order Detail, Order, Customer, Staff, Role and Payment.

The Location Table which stored the location of the Warehouse was connected to the Warehouse Table which stored the Products. The Warehouse Table connected to Supplier Table which was then connected to the Product Table. The use of a supplier table between warehouse and the product table was to ease this part of the process in the supply chain. The Category table stored the information about the type of products in the Inventory. The Order Detail contain info about the order and the Order Table contained info about the total order. The Customer, Staff, Role and the Payment Table are rather self-explanatory. The Staff ID in the customer Table is the id of the staff member which dealt with the customer. This marks the end of Component 2.

Moving on to Component 5, the tables we created for the ERD were brought to life in this part. We made a python script which generated data for our tables along with some fixed assumptions.

```
[241]: import pandas as pd
import random
from datetime import datetime, timedelta
from numpy.random import choice
import numpy as np

[273]: import datetime
import random

def main(config):
    random.seed(config["seed"])
    some_date = random_date()
    return some_date

def random_date():
    earliest = datetime.date(2019,1,1)
    latest = datetime.date(2020,1,1)
    delta = latest - earliest
    int_delta = (delta.days * 24 * 60 * 60) + delta.seconds
    random_second = random.randrange(int_delta)
    return earliest + datetime.timedelta(seconds = random_second)

[319]: Product=[]
data=[]
status=["Available","Out of Stock"]
name=["Lays","Pepsi","Lux","Coca Cola", "Cocomo","Mountain Dew","Surf Excel","Soooper", "Tapal Danedar", "Miranda", "Slanty", "Pearls","Lifebouy","Safe"]
for i in range(len(name)):
    productid = (12345+i)
    categoryid=(1000,1005)
    supplierid=i+100
    warehouseid=random.randint(111,112)
    pname = name[random.randint(0,len(name)-1)]
    weight= random.randint(50,900)
    price=random.randint(15,1001)
    status = choice(status, 1, p=[0.7,0.3])[0]
    data.append([productid, categoryid, supplierid, warehouseid,pname,weight,price,status])
df = pd.DataFrame(data,columns = ['Product ID', 'Category ID', 'Supplier ID', 'Warehouse ID', 'Product Name','Weight','Price','Status'])
pricelist=df[["Price"]].to_list()
productidlist=df[["Product ID"]].to_list()

[320]: df.to_csv('Product.csv',index=False)
```

Here we hardcoded some Products Name along with their status. Then we generated a huge amount of data randomly. The data was then saved into a CSV File. This is how most of generation of data was done.

```
Category = ["Sanitation","Biscuit","Soft Drinks","Tea","Chips","Ice Cream","Oil"]
description=["Personal hygiene products","Crackers, hard sweet and semi-sweet biscuits","A soft drink is a drink that usually contains carbonated water","An oil is any nonpolar chemical substance that...","Ice cream is a sweetened frozen food typically..."]
data = []
count=1000
for x in range(len(Category)):
    categoryname=Category[x]
    categoryid=count
    descriptionn=choice(description,1)[0]
    count=count+1
    data.append([categoryid,categoryname,descriptionn])
df2 = pd.DataFrame(data,columns = ['Category ID','Category Name', 'Category Description'])
df2

Category ID  Category Name  Category Description
0           1000    Sanitation  An oil is any nonpolar chemical substance that...
1           1001      Biscuit    Served as a snack, side dish, or appetizer
2           1002    Soft Drinks  Crackers, hard sweet and semi-sweet biscuits
3           1003         Tea    An oil is any nonpolar chemical substance that...
4           1004        Chips    Ice cream is a sweetened frozen food typically...
5           1005      Ice Cream    Served as a snack, side dish, or appetizer
6           1006         Oil    Ice cream is a sweetened frozen food typically...

df2.to_csv('Category.csv',index=False)
```

Category Data Generation

```

data=[]
Supplier = ["Pepsico","Peak Freans","LU","Uniliever","R8","Ismail Industries","CandyLand","Bashir Industries","Memon Mart","Habib","Meezan","Walls","O
Phone=["033334454","032432456","0324324234","023435435","032432116","035942213","0231234343","021314353","02442534521","032546547","0213012314","02121
email=["abc@pepsico.com","abc@peakfreans.com","abc@lu.com","abc@uniliever.com","abc@rb.com","abc@ismailindustries.com","abc@candyland.com","abc@bashiri
address= ["F-56 Block-2 , Kashmir Road","F-57 Block-3 , Kashmir Road","A-23 Block-5 , Kashmir Road","B-46 Block-2 , Kashmir Road","B-234 Meezan Bank R
for x in range(len(Supplier)):
    supid=random.randint(100,112)
    Supname=Supplier[x]
    Phonee=Phone[x]
    email=email[x]
    addressx=address[x]
    fax=random.randint(100123,142456)
    data.append([supid,Supname,Phonee,email,addressx,fax])
df3 = pd.DataFrame(data,columns = ["Supplier ID", 'Supplier Name', 'Supplier Phone', 'Supplier Email', 'Supplier Address','Supplier Fax number'])
supplieridlist=df3["Supplier ID"].to_list()

df3.to_csv('Supplier.csv',index=False)

```

Supplier Data Generation

```

fname = ["Inam","Zain","Saadat","Hasaham","Fatima","Haroon","Osama","Fahad","Amna","Ammara","Zubair","Sadiq","Aslam","Gohar","Qasim","Ayesha","Nazneen
lname=["Raza","Ali","Ouj","Raza","Ali","Raza","Raza","Raza","Raza","Raza","Ali", "B.", "A.", "C.", "Y.", "W.", "D.", "P.", "P.", "AB.", "P.", "P.", "A.", "A.", "A.", "A.",
Phonecus=["00324234234","0023435435","0032432116","0035942213","00231234343","0021314353","002442534521","0032546547","00213012314","00212142145","002
data=[]
letters = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z",]
emailcus=[]
for i in range(50):
    random1 = random.choice(letters)
    emailcus.append((random1+"@gmail.com"))
addresscus= ["B-134 Meezan Bank Road","C-210 Tipu Sultan","D-110, New Tipu Sultan","DS-110, Korangi","DS-123, Korangi Creek","DS-12, Defence","Y-124,I

for i in range(50):
    first=fname[i]
    last=lname[i]
    customerphone=Phonecus[i]
    customerid=random.randint(1100012,1100085)
    cusaddress=addresscus[i]
    email=emailcus[i]
    staffid=random.randint(222,522)
    data.append([customerid,first,last,customerphone,cusaddress,email,staffid])
df4 = pd.DataFrame(data,columns = ["Customer ID", 'Customer First Name', 'Customer Last Name', 'Customer Phone', 'Customer Address','Customer Email',"S
df4

```

Customer Data Generation

```

Order=[]
data=[]
for i in range(50000):
    orderid=random.randint(112345,195555)
    dateorder=random_date()
    pnames = choice(name, 4)
    customerid=random.randint(1100012,1100065)
    Order.append([orderid,dateorder,customerid])
df5 = pd.DataFrame(Order,columns = ["Order ID", 'Order Date', 'Customer ID'])
orderlist=df5["Order ID"].to_list()
orderdatelist=df5["Order Date"].to_list()
df5

```

Order Data Generation

```

orderdetails=[]
pricelist
size=[3,4]
Discount=[]
for i in range(10000):
    detailid=random.randint(123,999)
    orderid=orderlist[i]
    productid=productidlist[random.randint(0,len(productidlist)-1)]
    sizes=size[random.randint(0,len(size)-1)]
    pnamesx = choice(name, sizes)
    price= choice(pricelist, sizes)
    orderdate=choice(orderdatelist,1)[0]
    Billnumber=random.randint(55555,666666)
    Total=sum(price)
    orderdetails.append([detailid,orderid,productid,size,str(pnames),str(price),orderdate,Billnumber>Total])
df6 = pd.DataFrame(orderdetails,columns = ["Detail ID",'Order ID', 'Product ID','Size',"Product Name","Price","Order Date","Bill No","Total"])

```

orderdetails

Order Details Generation

```

data=[]
warehousename=["Warehouse1","Warehouse11","Warehouse111","Warehouse1123","Warehouse1123","Warehouse23","Warehouse223","Warehouse256","Warehouse246"]
for i in range(len(warehousename)):
    warehouseid=i+1
    ware=warehousename[i]
    locationid=random.randint(7777,8888)
    supid=supplieridlist[random.randint(0,len(supplieridlist)-1)]
    data.append([warehouseid,ware,locationid,supid])
df7 = pd.DataFrame(data,columns = ["Warehouse ID",'Warehouse Name', 'Location ID','Supplier ID'])
locationidlist=df7['Location ID'].to_list()

```

Warehouse Data Generation

```

Data=[]
address
City=["Karachi","Lahore","Islamabad","Faislabad","Dera Ghazi Khan","Sukkur","Quetta","Peshawar","Hunza","Sialkot","Larkana","Rahim Yar Khan","Dadu","M
Province=["Sindh","Balochistan","KPK","Punjab"]
for i in range(5000):
    locationid=choice(locationidlist,1)[0]
    Address=choice(address,1)[0]
    cityy=choice(City,1)[0]
    vince=choice(Province,1)[0]
    Postalcode=random.randint(54000,55000)
    Data.append([locationid,Address,cityy,vince,Postalcode])
df8= pd.DataFrame(Data,columns = ["Location ID", "Address","City","Province","Postal Code"])

```

df8

	Location ID	Address	City	Province	Postal Code
0	8853	B-134 Meezan Bank Road	Multan	Balochistan	54052
1	8746	B-134 Meezan Bank Road	Sukkur	KPK	54941
2	8859	F-56 Block-2 , Kashmir Road	Larkana	Punjab	54564
3	8853	B-234 Meezan Bank Road	Topi	Sindh	54615
4	8385	F-56 Block-2 , Kashmir Road	Islamabad	Sindh	54331
...

Location Data Generation

The whole process was to create DataFrames in Python, insert random data with a structure in it and convert those dataframes into actual CSV files.

This is how synthetic data was added into our tables and this marks the end of Component 5.

Now the final Component of our track 2 which is Component 6, we generated tables over Hive to facilitate dimensional modeling requirements. Then, we imported our data into these tables and executed them to get our results.

The screenshot shows the Hive Editor interface with the 'Query Editor' tab selected. The query editor contains the following SQL query:

```
1 Select * from customer where customerlastname = "Raza"
```

Below the query editor, there are buttons for 'Execute', 'Save as...', 'Explain', 'Format', and 'New query'. The 'Results' tab is selected, displaying the following table:

	customer.customerid	customer.customerfirstname	customer.customerlastname	customer.customerphone	customer.customeraddress
1	1100024	Inam	Raza	00324234234	B-134 Meezan Bank Road
2	1100026	Hasaham	Raza	0035942213	"DS-110
3	1100070	Haroon	Raza	0021314353	"DS-12
4	1100044	Osama	Raza	002442534521	"Y-124
5	1100066	Fahad	Raza	0032546547	"AF-56 Block-2
6	1100083	Amna	Raza	00213012314	"F-56 Block-2

The screenshot shows the Hive Editor interface with the 'Query Editor' tab selected. The query editor contains the following SQL query:

```
1 Select * from supplier where supplieremail = "abc@lu.com"
```

Below the query editor, there are buttons for 'Execute', 'Save as...', 'Explain', 'Format', and 'New query'. The 'Results' tab is selected, displaying the following table:

	supplier.supplierid	supplier.suppliername	supplier.supplierphone	supplier.supplieremail	supplier.supplieraddress	supplier.suppliercity
1	103	LU	0324234234	abc@lu.com	"A-23 Block-5	NULL

The screenshot shows the Hive Editor interface. The top navigation bar includes 'Query Editors', 'Data Browsers', 'Workflows', 'Search', and 'Security'. The left sidebar shows a tree view of tables: 'default' (7 tables), 'category', 'customer', 'location' (address, city, locationid, postalcode, province), 'orders', 'product', 'supplier', and 'warehouse' (locationid, supplierid, warehouseid, warehousename). The main query editor contains the following SQL query:

```
1 Select locationid,city from location where city= "Karachi"
```

Below the query editor are buttons for 'Execute', 'Save as...', 'Explain', 'Format', 'or create a', and 'New query'. The 'Results' tab is selected, showing a table with two columns: 'locationid' and 'city'.

	locationid	city
1	8851	Karachi
2	7941	Karachi
3	8853	Karachi
4	8564	Karachi
5	8385	Karachi
6	8385	Karachi

The screenshot shows the Hive Editor interface. The top navigation bar includes 'Query Editors', 'Data Browsers', 'Workflows', 'Search', and 'Security'. The left sidebar shows a tree view of tables: 'default' (7 tables), 'category', 'customer' (customeraddress, customeremail, customerfirstname, customerid, customerlastname, customerphone, staffid), 'location', 'orders', and 'product' (categoryid, price, productname, productid, status, supplierid). The main query editor contains the following SQL query:

```
1 Select productid,price from product where price > 500
```

Below the query editor are buttons for 'Execute', 'Save as...', 'Explain', 'Format', 'or create a', and 'New query'. The 'Results' tab is selected, showing a table with two columns: 'productid' and 'price'.

	productid	price
1	12345	636
2	12348	562
3	12355	803
4	12357	742
5	12359	728

This marks the end of our Component 6 and Our Project.

This was one of the most important Projects in our course and we felt we learnt a lot while attempting it.