School of Information Studies

PROJECT

DATA ADMINISTRATION CONCEPTS AND DATABASE MANAGEMENT

IST – 659

FALL 2022

WHOLESALE INVENTORY MANAGEMENT SYSTEM

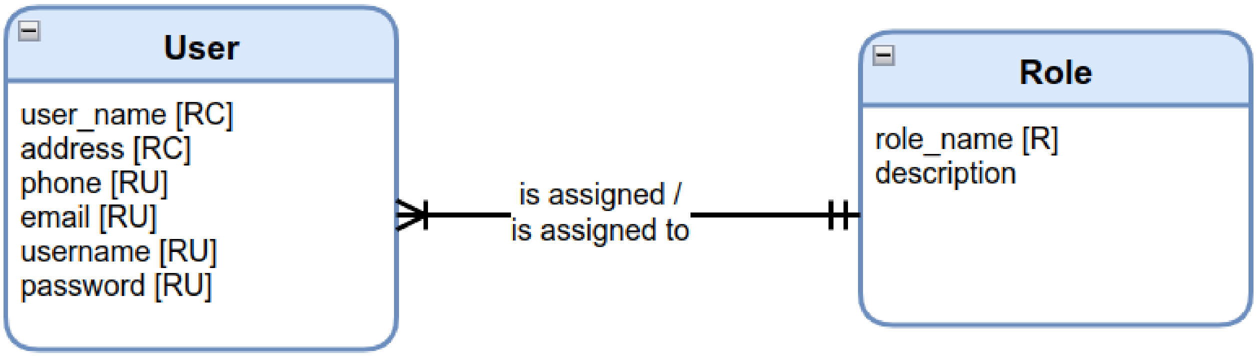
PREPARED BY : **HENDI KUSHTA**

Introduction

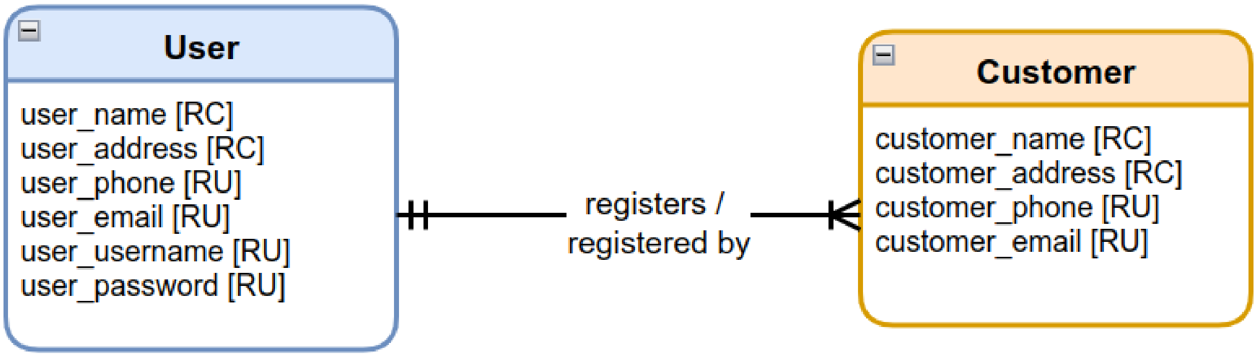
In order to manage the suppliers, products, item stock, purchase orders, and customer orders in a wholesale business, I will design a database schema for the inventory management system.
Inventory management systems are frequently used by manufacturing and retail companies. This database can be used, for example, to manage wholesale inventories more effectively and cut down on product waste in supply chains.
Software that assists in tracking a company's inventory and sales is known as an inventory management system. It supports employees by providing accurate inventory numbers and keeping products organized. This assists a business in managing its inventory, completing orders, and monitoring overall output or sales. Inventory management can be more effective by using an electronic system, which requires little training.
Systems for managing inventory help businesses cut expenses by avoiding overstocking. A business can guarantee sales orders for customers and avoid back-orders when it has an accurate inventory count.
Typically, an inventory system has four basic elements: products, purchases, orders, and suppliers. Order alerts can be set to trigger when inventory levels fall below custom-defined minimum levels.
By the end of this project, I will build an application using power apps for the inventory management. I will also use Power BI Desktop to build reports for the items in stock, sold, customer orders etc.

Business Rules

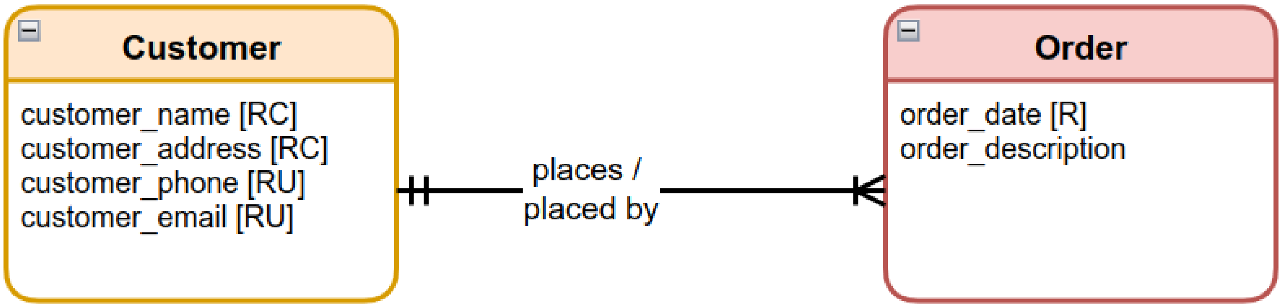
1 - The app will have a user. Each user will have their information like first name, last name, address, phone, email. As a user/staff I need to have my own username and password to access the system. Each user will have a role in the company like manager, employee etc.



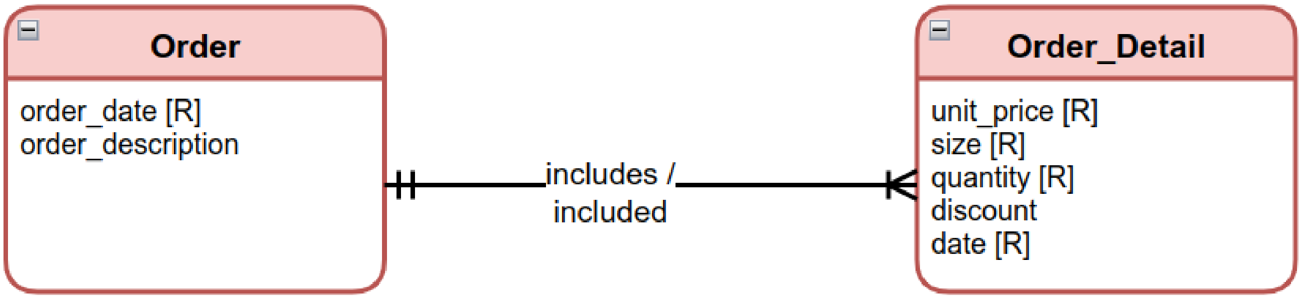
2 - As a stuff person, I have to register customers. Each customer mus have a name, address, phone and email address.



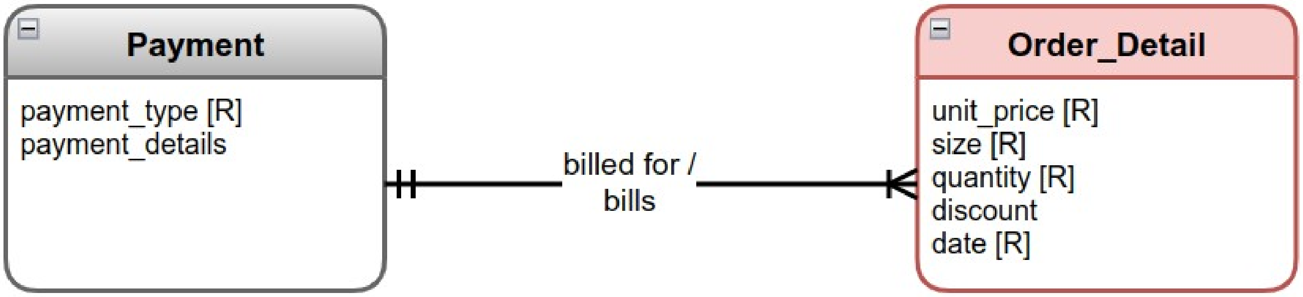
3 – Each customer make orders. Each order will have a date of order, order description.



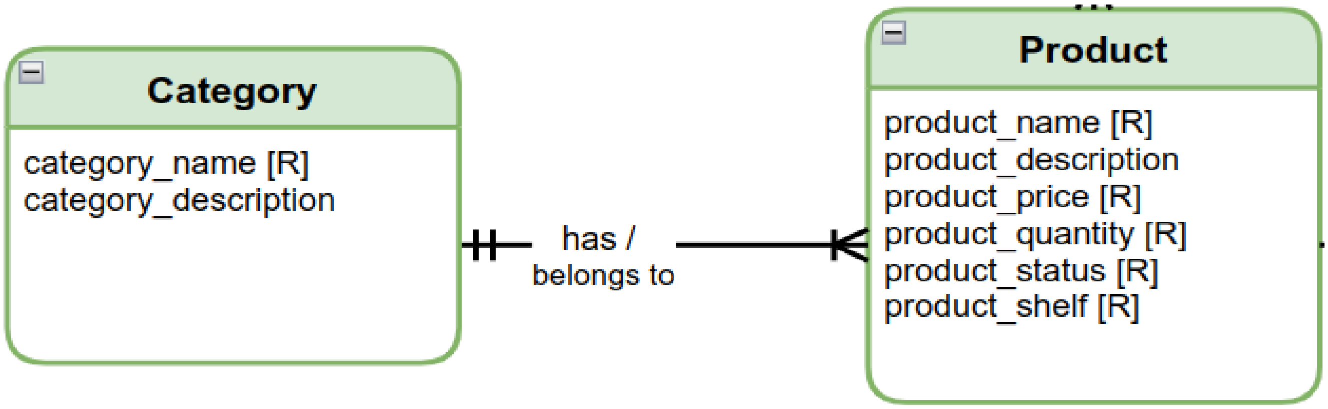
4 – Each order has order_details which includes price, quantity, size, date, discount.



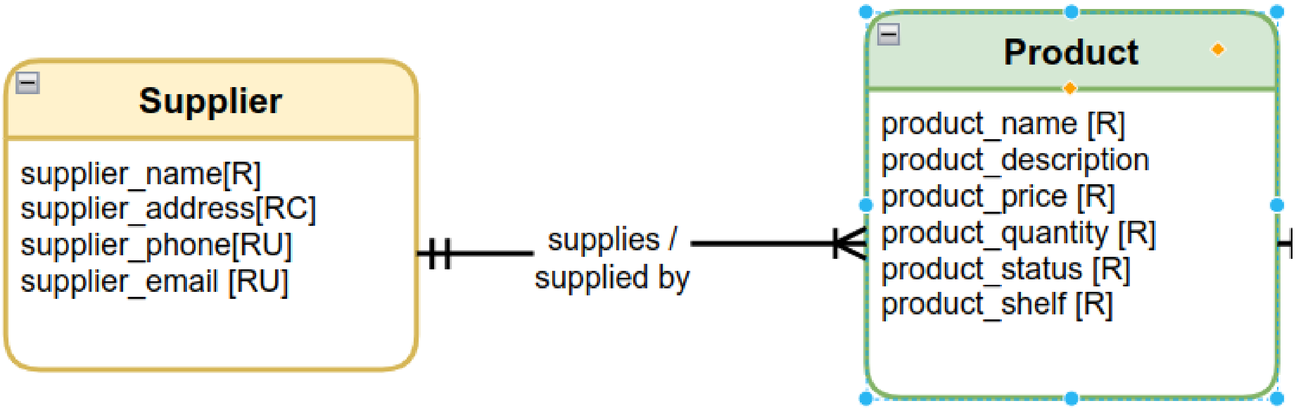
5 – For each order there should be a payment. Payments might be of different types, cash, card etc. It will also include description or details.



6 – Each product belongs to a category. Products will have a name, description, unit, price, quantity, which shelves are found and a product status, found or not. Categories will have a name, and a description. - **Chintan Patel**



7 – Products are supplied by vendors. Each vendors will have a name, address, phone, fax, email. - **Chintan Patel**

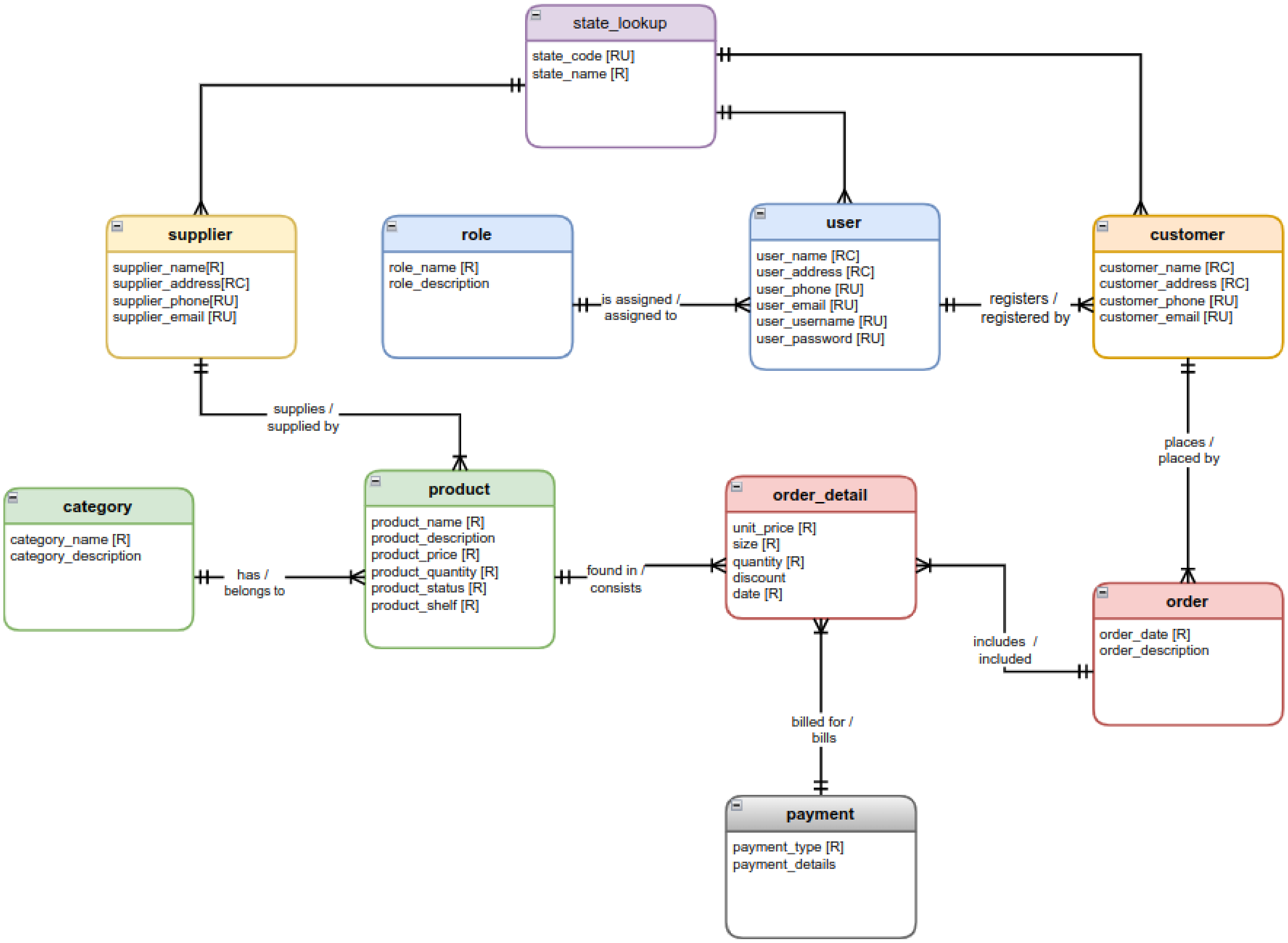


User Stories

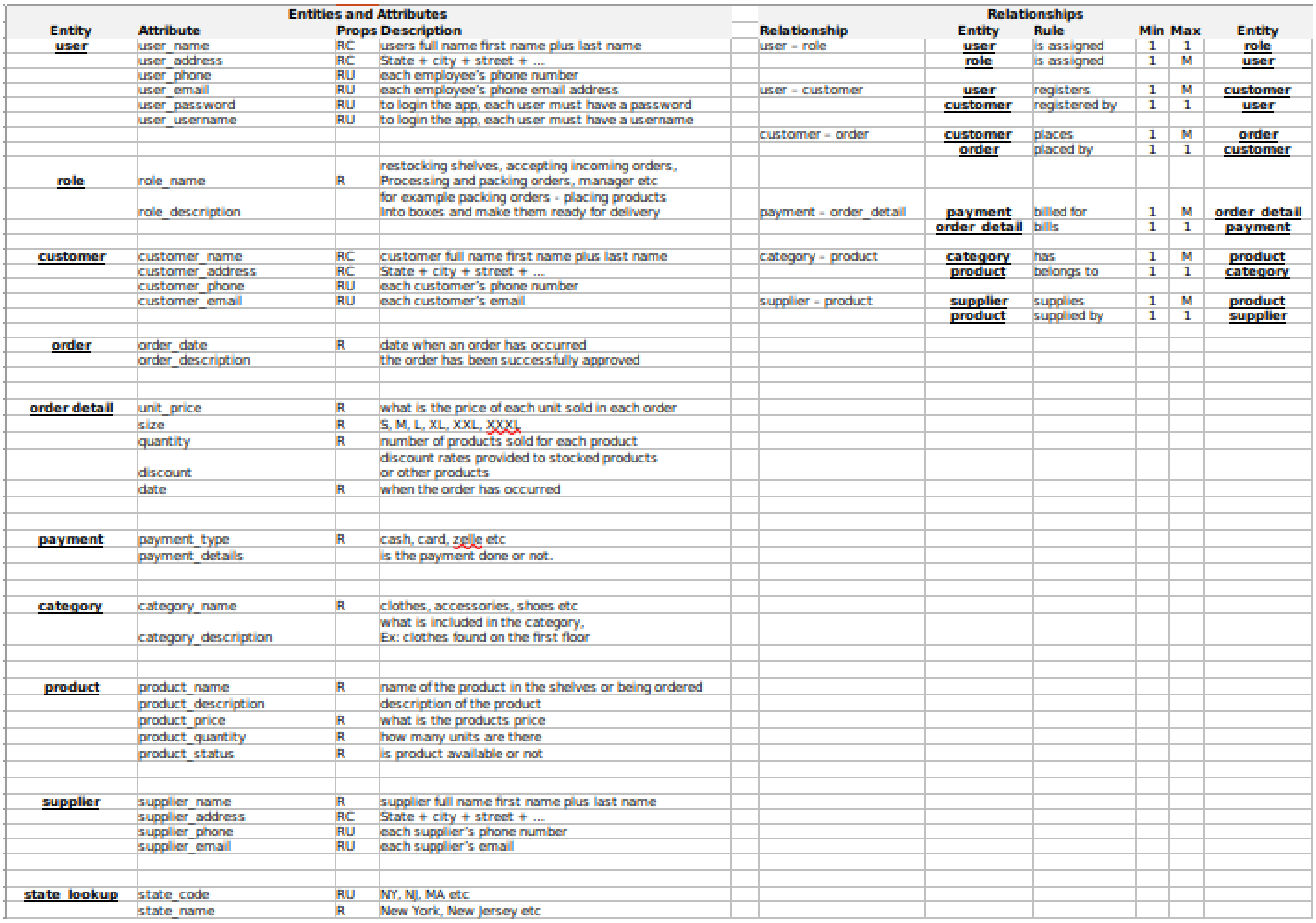
1- A customer buys a product. The customer needs to provide his information like name and surname, state etc. When he buys something, an order its details are created which will show quantity, unit price etc. A column will be created which will find the total amount of money a person has spent for each product and the payment type the paid.

To create this user story I have built 2 views and from these 2 views I have created 2 graphs also shown in the power app.

Conceptual Model

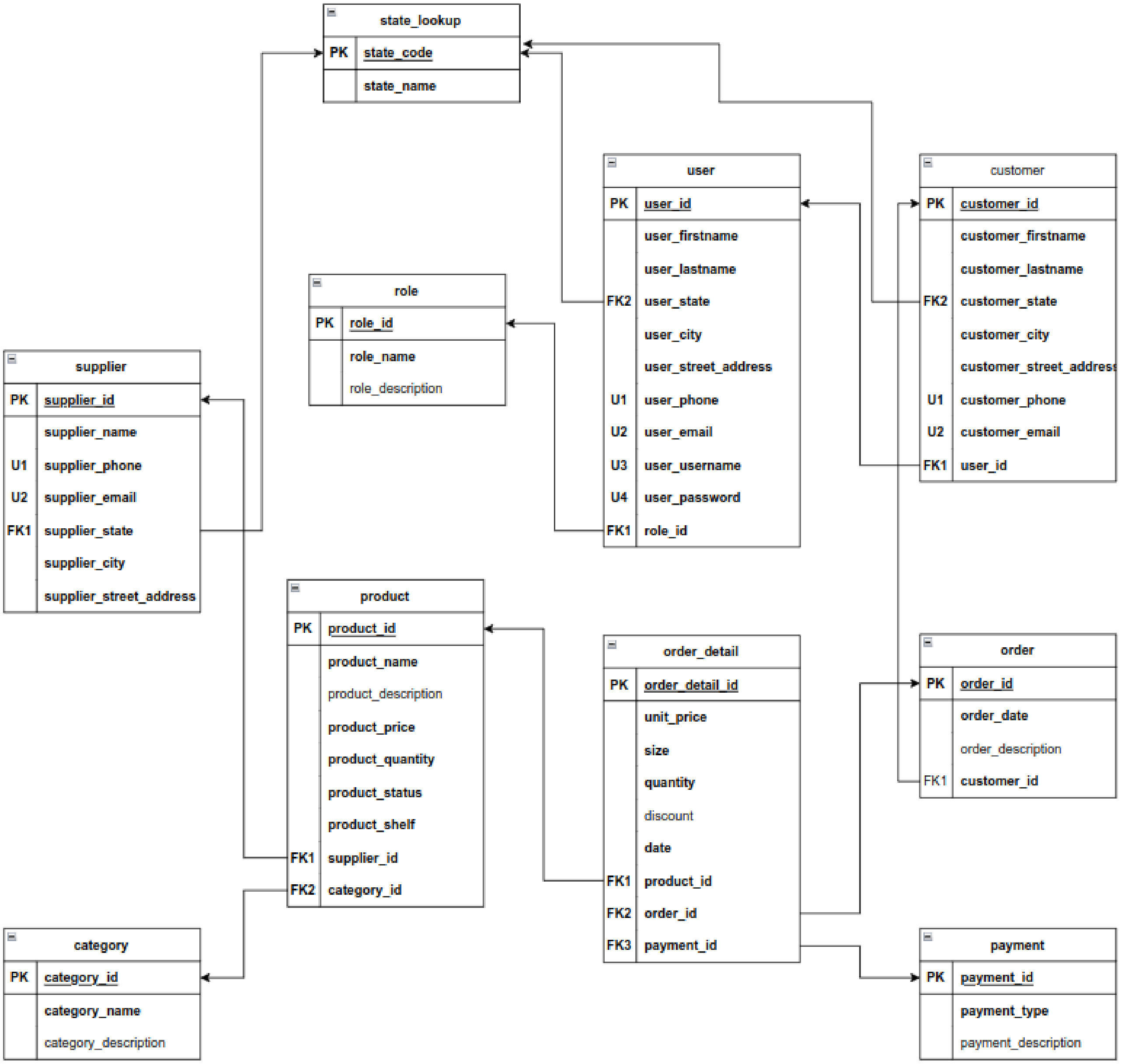


E-R Requirements



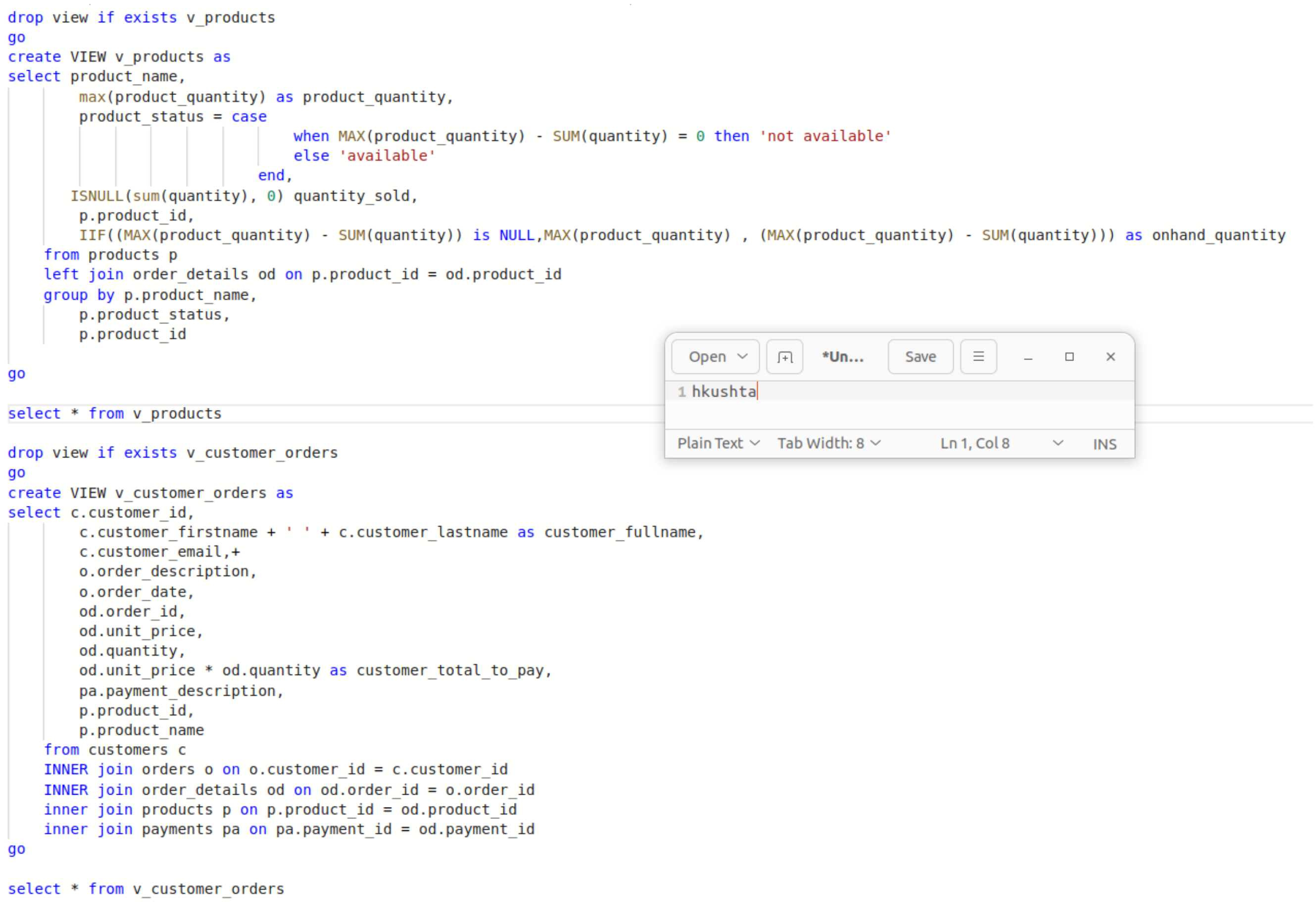
Entities and Attributes				Relationships					
Entity	Attribute	Props	Description	Relationship	Entity	Rule	Min	Max	Entity
user	user_name	RC	users full name first name plus last name	user - role	**user**	is assigned	1	1	**role**
	user_address	RC	State + city + street + ...		**role**	is assigned	1	M	**user**
	user_phone	RU	each employee's phone number						
	user_email	RU	each employee's phone email address	user - customer	**user**	registers	1	M	**customer**
	user_password	RU	to login the app, each user must have a password		**customer**	registered by	1	1	**user**
	user_username	RU	to login the app, each user must have a username						
				customer - order	**customer**	places	1	M	**order**
					order	placed by	1	1	**customer**
role	role_name	R	restocking shelves, accepting incoming orders, Processing and packing orders, manager etc						
	role_description		for example packing orders - placing products Into boxes and make them ready for delivery	payment - order_detail	**payment**	billed for	1	M	**order_detail**
					order_detail	bills	1	1	**payment**
customer	customer_name	RC	customer full name first name plus last name	category - product	**category**	has	1	M	**product**
	customer_address	RC	State + city + street + ...		**product**	belongs to	1	1	**category**
	customer_phone	RU	each customer's phone number						
	customer_email	RU	each customer's email	supplier - product	**supplier**	supplies	1	M	**product**
					product	supplied by	1	1	**supplier**
order	order_date	R	date when an order has occurred						
	order_description		the order has been successfully approved						
order detail	unit_price	R	what is the price of each unit sold in each order						
	size	R	S, M, L, XL, XXL, XXXL						
	quantity	R	number of products sold for each product						
	discount		discount rates provided to stocked products or other products						
	date	R	when the order has occurred						
payment	payment_type	R	cash, card, zelle etc						
	payment_details		is the payment done or not.						
category	category_name	R	clothes, accessories, shoes etc						
	category_description		what is included in the category, Ex: clothes found on the first floor						
product	product_name	R	name of the product in the shelves or being ordered						
	product_description		description of the product						
	product_price	R	what is the products price						
	product_quantity	R	how many units are there						
	product_status	R	is product available or not						
supplier	supplier_name	R	supplier full name first name plus last name						
	supplier_address	RC	State + city + street + ...						
	supplier_phone	RU	each supplier's phone number						
	supplier_email	RU	each supplier's email						
state_lookup	state_code	RU	NY, NJ, MA etc						
	state_name	R	New York, New Jersey etc						

Logical Model



Data logics

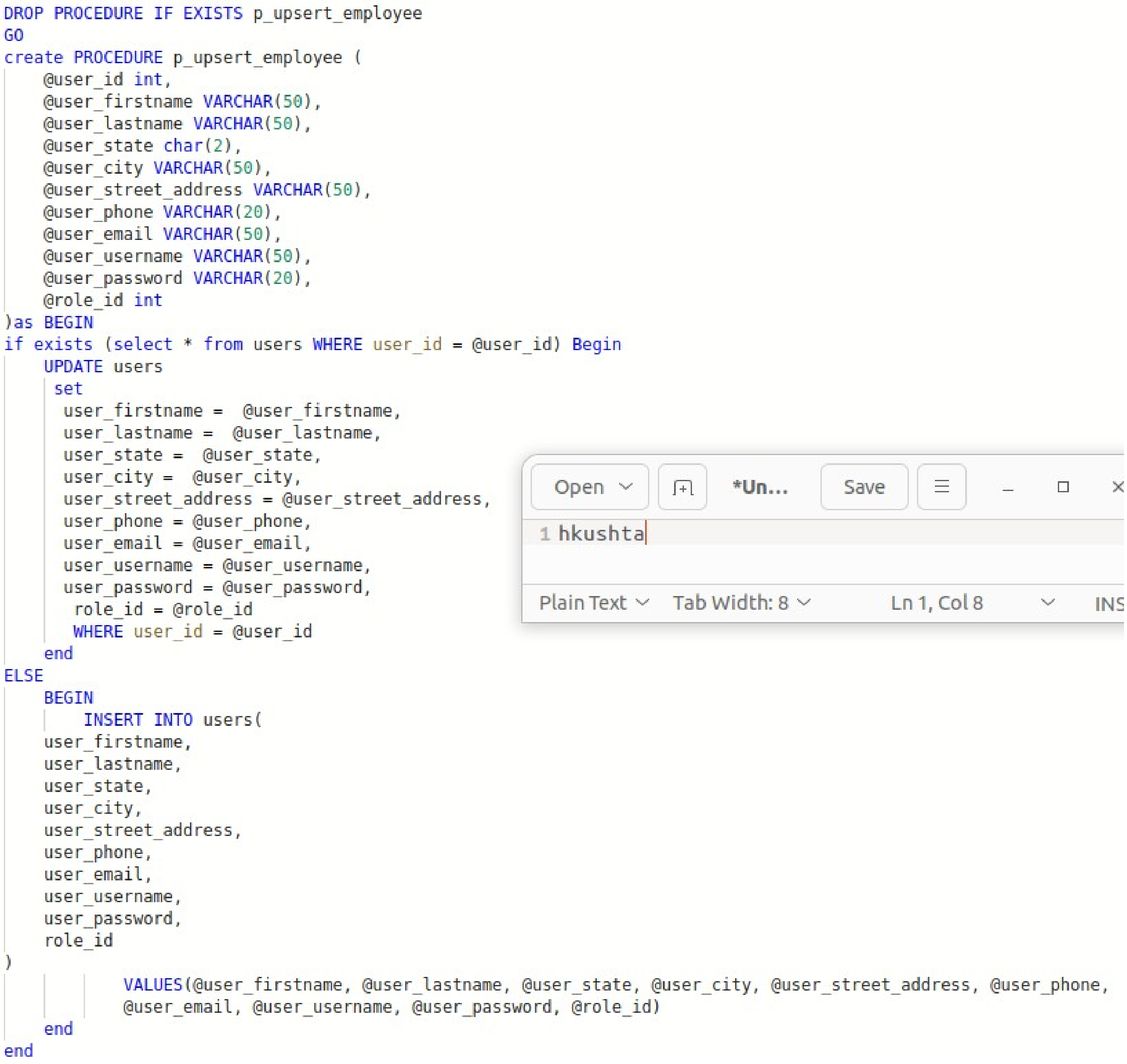
Have created 2 views to build the proper graphs for the data analysis as shown below.



```
drop view if exists v_products
go
create VIEW v_products as
select product_name,
        max(product_quantity) as product_quantity,
        product_status = case
                                when MAX(product_quantity) - SUM(quantity) = 0 then 'not available'
                                else 'available'
                            end,
       ISNULL(sum(quantity), 0) quantity_sold,
        p.product_id,
        IIF((MAX(product_quantity) - SUM(quantity)) is NULL,MAX(product_quantity) , (MAX(product_quantity) - SUM(quantity))) as onhand_quantity
    from products p
    left join order_details od on p.product_id = od.product_id
    group by p.product_name,
        p.product_status,
        p.product_id

go

select * from v_products

drop view if exists v_customer_orders
go
create VIEW v_customer_orders as
select c.customer_id,
        c.customer_firstname + ' ' + c.customer_lastname as customer_fullname,
        c.customer_email,+
        o.order_description,
        o.order_date,
        od.order_id,
        od.unit_price,
        od.quantity,
        od.unit_price * od.quantity as customer_total_to_pay,
        pa.payment_description,
        p.product_id,
        p.product_name
    from customers c
    INNER join orders o on o.customer_id = c.customer_id
    INNER join order_details od on od.order_id = o.order_id
    inner join products p on p.product_id = od.product_id
    inner join payments pa on pa.payment_id = od.payment_id
go

select * from v_customer_orders
```

hkushta

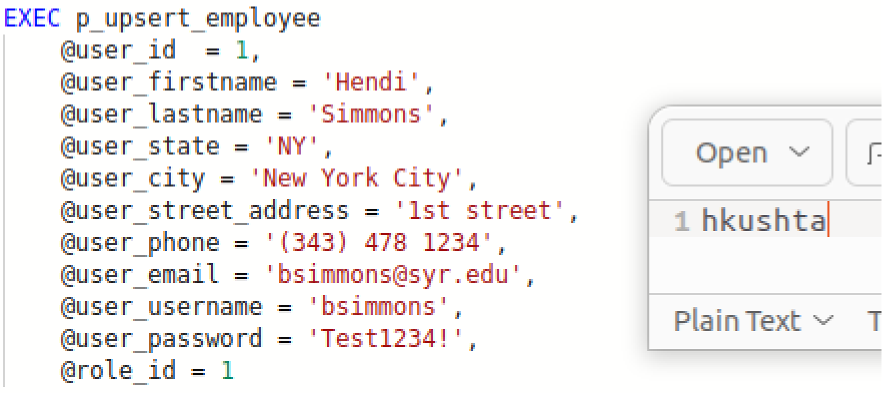
I have also created a procedure when to update an employee if it exists and create a new one if it doesn’t.



```
DROP PROCEDURE IF EXISTS p_upsert_employee
GO
create PROCEDURE p_upsert_employee (
    @user_id int,
    @user_firstname VARCHAR(50),
    @user_lastname VARCHAR(50),
    @user_state char(2),
    @user_city VARCHAR(50),
    @user_street_address VARCHAR(50),
    @user_phone VARCHAR(20),
    @user_email VARCHAR(50),
    @user_username VARCHAR(50),
    @user_password VARCHAR(20),
    @role_id int
)as BEGIN
if exists (select * from users WHERE user_id = @user_id) Begin
    UPDATE users
     set
      user_firstname =  @user_firstname,
      user_lastname =  @user_lastname,
      user_state =  @user_state,
      user_city =  @user_city,
      user_street_address = @user_street_address,
      user_phone = @user_phone,
      user_email = @user_email,
      user_username = @user_username,
      user_password = @user_password,
       role_id = @role_id
       WHERE user_id = @user_id
    end
ELSE
    BEGIN
        INSERT INTO users(
    user_firstname,
    user_lastname,
    user_state,
    user_city,
    user_street_address,
    user_phone,
    user_email,
    user_username,
    user_password,
    role_id
)
            VALUES(@user_firstname, @user_lastname, @user_state, @user_city, @user_street_address, @user_phone,
            @user_email, @user_username, @user_password, @role_id)
    end
end
```

hkushta

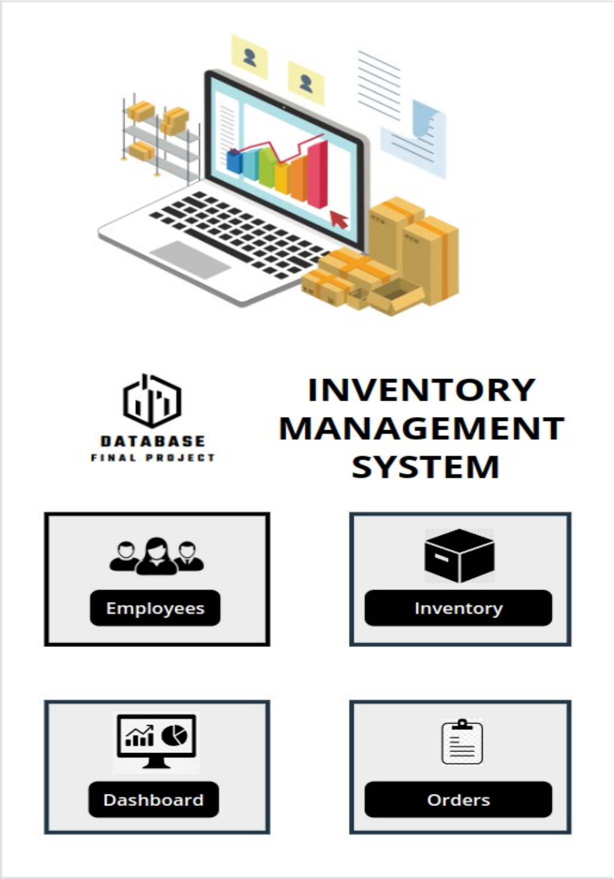
I have updated just the first name for the first user

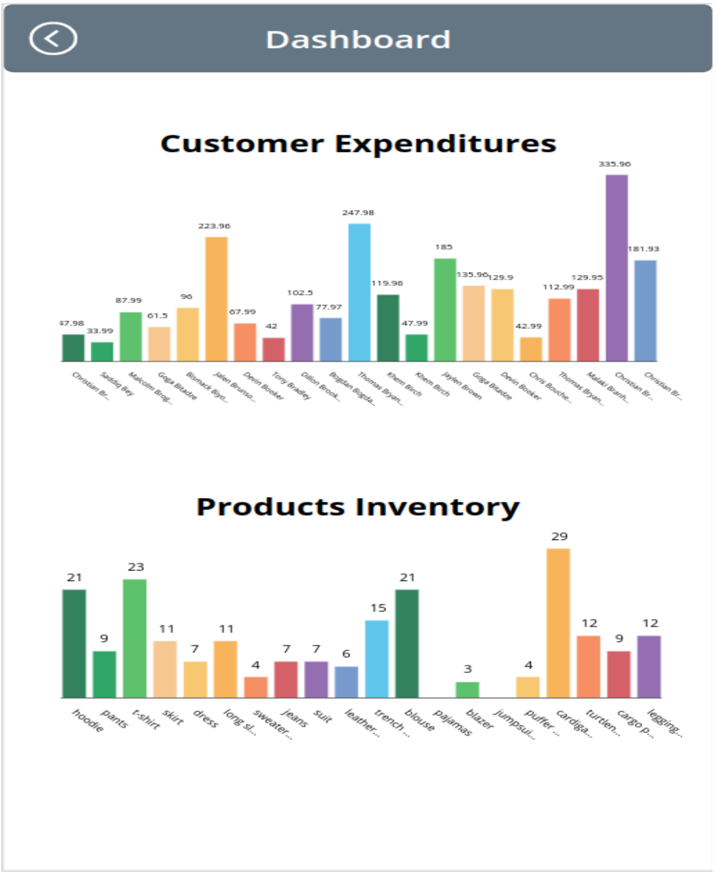


```
EXEC p_upsert_employee
    @user_id  = 1,
    @user_firstname = 'Hendi',
    @user_lastname = 'Simmons',
    @user_state = 'NY',
    @user_city = 'New York City',
    @user_street_address = '1st street',
    @user_phone = '(343) 478 1234',
    @user_email = 'bsimmons@syr.edu',
    @user_username = 'bsimmons',
    @user_password = 'Test1234!',
    @role_id = 1
```

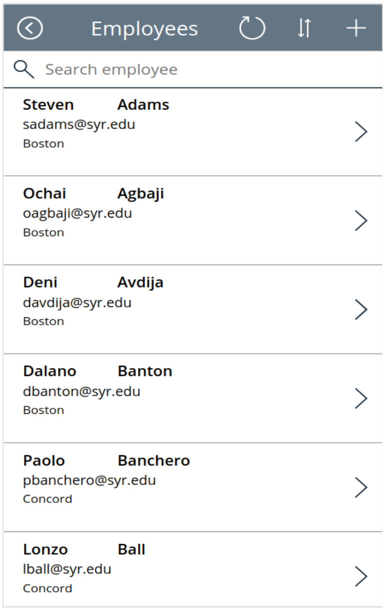
hkushta

Power App





When I click in the dashboard, we can see that there are 2 plots. This plots answer the questions, which customer has made the most purchases. And what is the inventory balance for each of the products. Done with the help of views I have created.



Steven Adams
sadams@syr.edu
Boston

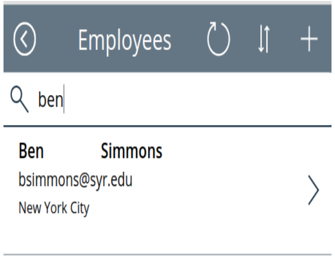
Ochai Agbaji
oagbaji@syr.edu
Boston

Deni Avdija
davdija@syr.edu
Boston

Dalano Banton
dbanton@syr.edu
Boston

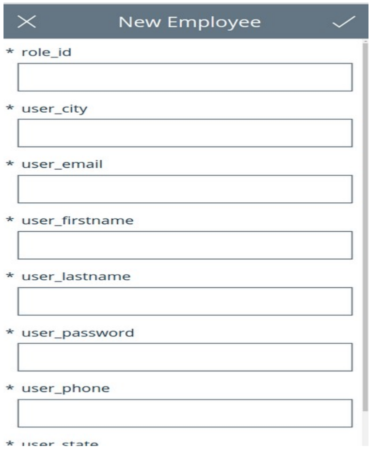
Paolo Banchero
pbanchero@syr.edu
Concord

Lonzo Ball
lball@syr.edu
Concord



ben

Ben Simmons
bsimmons@syr.edu
New York City



* role_id

* user_city

* user_email

* user_firstname

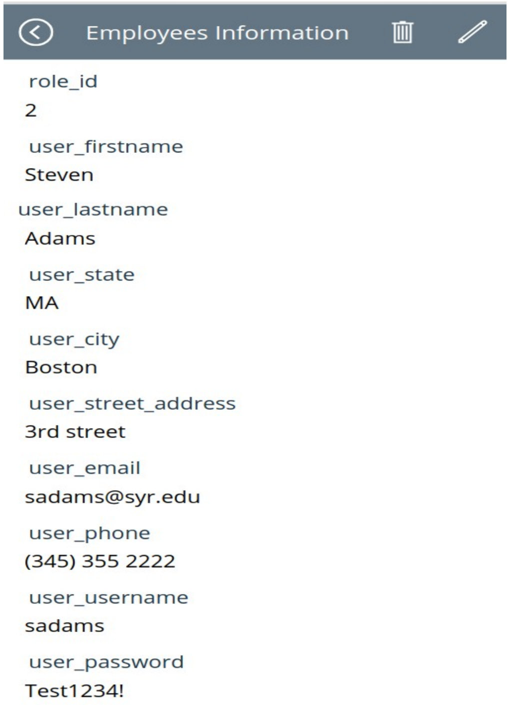
* user_lastname

* user_password

* user_phone

* user_state

In the above figures we can see all the employees, filter them and create a new one.
As I mentioned before I have created also a procedure to update the employee he/she is exsiting.



role_id
2

user_firstname
Steven

user_lastname
Adams

user_state
MA

user_city
Boston

user_street_address
3rd street

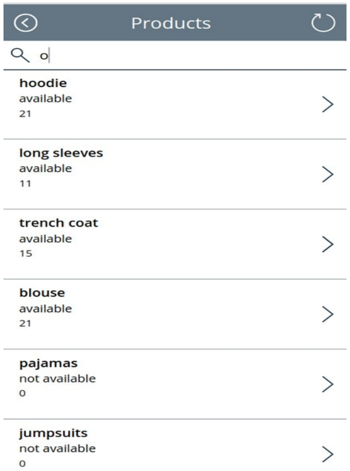
user_email
sadams@syr.edu

user_phone
(345) 355 2222

user_username
sadams

user_password
Test1234!

I have also used the created view to find and update if the products are not in the inventory anymore, they change the status from available to not available. Just like shown below. If its not available like in the example of pajamas, we can click + button to send an email to our supplier so they can bring more, but the window is not functional.



o

hoodie
available
21

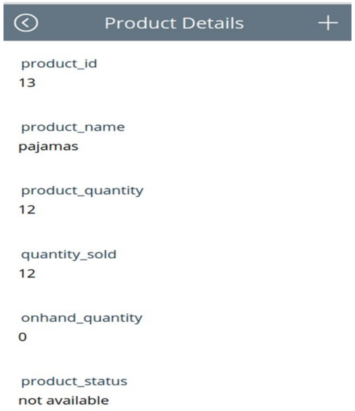
long sleeves
available
11

trench coat
available
15

blouse
available
21

pajamas
not available
0

jumpsuits
not available
0



product_id
13

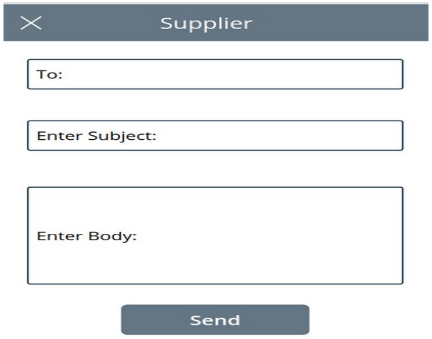
product_name
pajamas

product_quantity
12

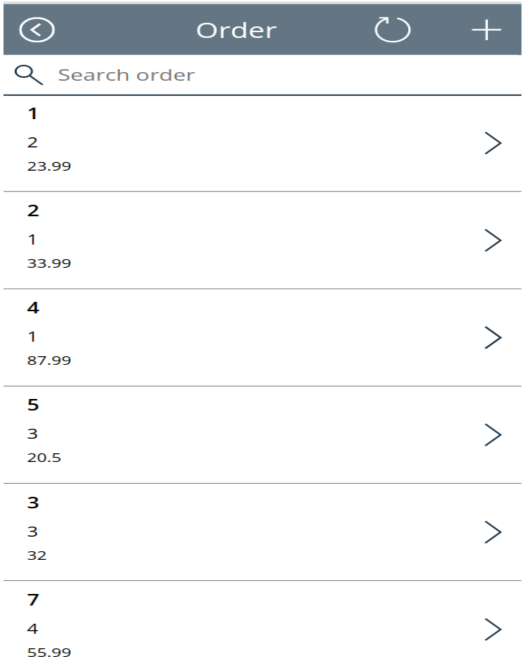
quantity_sold
12

onhand_quantity
0

product_status
not available



On the screen shot below you can find the window for orders and how to create a new order.



1
2
23.99

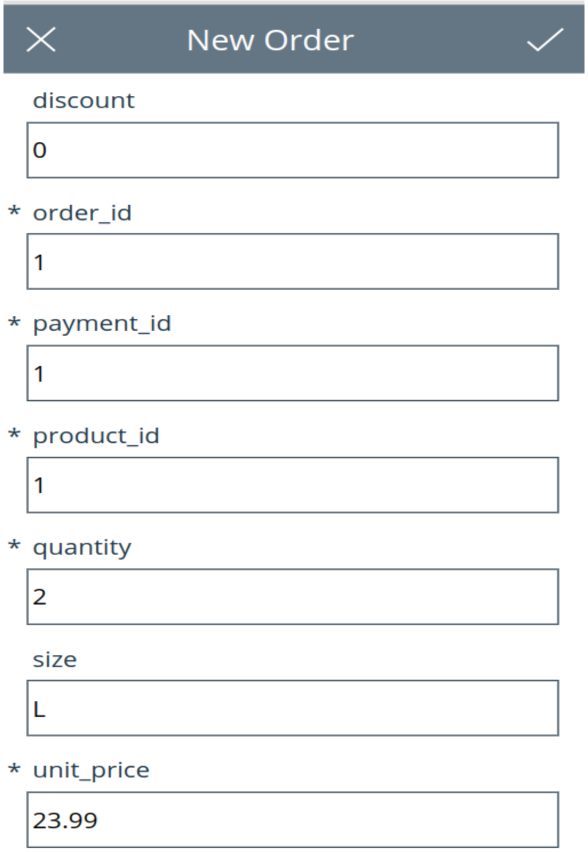
2
1
33.99

4
1
87.99

5
3
20.5

3
3
32

7
4
55.99



discount: 0

* order_id: 1

* payment_id: 1

* product_id: 1

* quantity: 2

size: L

* unit_price: 23.99