Part 1 )

* **SQL to check if a person is an employee along with matching if they are a Manger base on our role system of 1 for Sale Representive, 2 for manager, and 3 for CEO.** After locate the role\_title then next step would require a server-client to do a javascript to notify the manger when a product is out of stock.

Select p.FName, P.LName, r.role\_ID, r.Role\_Title from person p join employees e on p.person\_ID = e.person\_ID join role r on r.Role\_ID = e.role\_Id where r.Role\_ID =2;

A screenshot of a cell phone

Description automatically generated

* **Check if vegetable or Fruit QTY lower than 2000;**

Select vege\_ID, Vege\_Name, Vege\_qty, Vege\_Price from vegetables where vege\_qty <= 2000;

Select \* from fruits where fruit\_qty <= 2000;

A screenshot of a cell phone

Description automatically generated

* **Update Customer information (address) using same id of the Customer:**

//First locate the user want to change

Ex: locate customer name John and update his phone to 651-799-0102

1)

Select \* from person p where p.FName like '%Steven%' and p.LName like '%Stevenson%';

2)

Update Record set Phone = '(651)-799-0102' where person\_ID = 11231;

3)

Check if it update successful

Select \* from record where person\_ID =11231;A screenshot of a cell phone

Description automatically generated

**Locate Customer Card Number for a person**

Select p.Person\_type, p.FName, c.Customer\_ID, y.Card\_Type, y.Pay\_CardNum from person p join p\_customer c on p.Person\_ID = c.person\_ID join payment y on c.Customer\_ID = y.Customer\_ID ;

A screenshot of a cell phone

Description automatically generated

**Determine if a customer has pay for a membership**

Select p.FName, e.Member\_ID, c.Customer\_ID, e.Mem\_Pay\_status from person p join p\_customer c on p.Person\_ID = c.person\_ID join membership e on c.member\_ID = e.member\_ID;

A screenshot of a cell phone

Description automatically generated

**What type of fruits and vegetables store in the warehouse**

Select \* from vegetables;

Select \* from fruits;

A screenshot of a cell phone

Description automatically generated

**Relational Algebra:**

Select vege\_ID, Vege\_Name, Vege\_qty, Vege\_Price from vegetables where vege\_qty <= 2000;

Π vege\_ID, Vege\_Name, Vege\_qty, Vege-Price (σ veget\_qty <=2000 (vegetables)

Select p.FName, P.LName, r.role\_ID, r.Role\_Title from person p join employees e on p.person\_ID = e.person\_ID join role r on r.Role\_ID = e.role\_Id where r.Role\_ID =2;

Π p.FName, p.LName, r.role\_ID, r.Role\_Title (Person  Role)

Part 2:

**Suggest or comment for server-client**: database might have some slow performance since there many tables create in our database. A website for allow user to input information such as customer information, employee information would be better to combine some of the table together for better performance when load or create new data into the database. There might be hard or easy depend on the skills of the server person to do a JavaScript or small programing script to storage information user type in or connect user to employee on either a view page for employee or customer once login the website.

**Suggestion or comment for myself:**

Do more research about a warehouse database or gain more experience in a warehouse database might help me understand more about what part need to be in the database such as product can it store both fruit and vegetable or what other thing that was important that a warehouse need to have in their database that I missing. Create less table to do a less join statement when writing a SQL statement might be faster and easier to grab data from table.