Name: Sarvesh Kaushik LAB 10

Instructor: Luke Papademas

Consider the following database schema for the supply chain management system.

**Suppliers Table:**

CREATE TABLE SUPPLIERS

(SUPPLIER\_ID INT PRIMARY KEY,

SUPPLIER\_NAME VARCHAR(25),

LOCATION\_CODE INT,

CONTACT\_PERSON VARCHAR(25),

OTHER\_DETAILS VARCHAR(50));

insert into suppliers values (15, 'Mary Supplies', 60645,'Johnny','N/A');

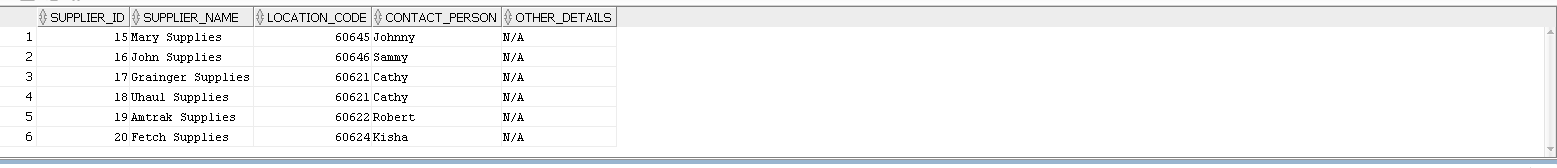
insert into suppliers values (16, 'John Supplies', 60646,'Sammy','N/A');

insert into suppliers values (17, 'Grainger Supplies', 60621,'Cathy','N/A');

insert into suppliers values (18, 'Uhaul Supplies', 60621,'Cathy','N/A');

insert into suppliers values (19, 'Amtrak Supplies', 60622,'Robert','N/A');

insert into suppliers values (20, 'Fetch Supplies', 60624,'Kisha','N/A');



**Products Table:**

Creating Products for supply chain management

CREATE TABLE PRODUCTS

(PRODUCT\_ID INT PRIMARY KEY,

PRODUCT\_NAME VARCHAR2(25),

PRODUCT\_DESCRIPTION VARCHAR2(30),

PRICE NUMBER,

SUPPLIER\_ID INT CONSTRAINT FK\_SUPPLIER\_ID REFERENCES SUPPLIERS (SUPPLIER\_ID),

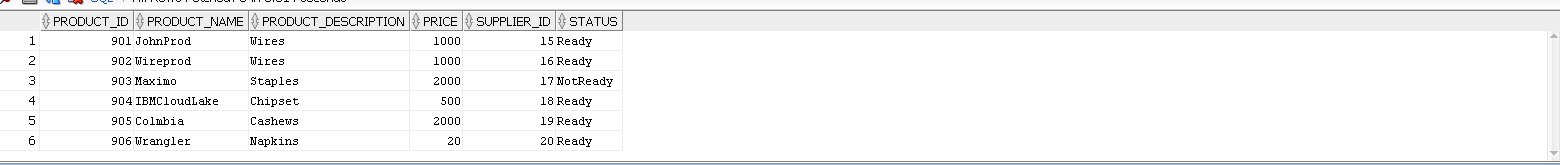
STATUS VARCHAR2(20));

insert into products values(901, 'JohnProd', 'Wires', 1000, 15, 'Ready');

insert into products values(902, 'Wireprod', 'Wires', 1000, 16, 'Ready');

insert into products values(903, 'Maximo', 'Staples', 2000, 17, 'NotReady');

insert into products values(903, 'Wrangler', 'Napkins', 20, 20, 'Ready');



insert into customer values(619, 'Dr Mysterio', 17, 2, 'Pro Wrestler');

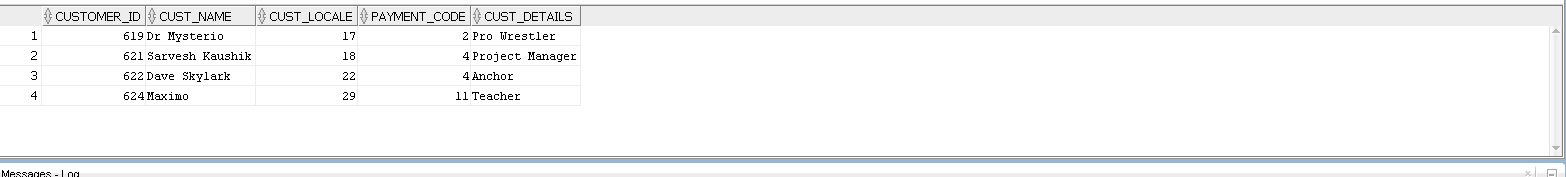
insert into customer values(620, 'John Hopkins', 18, 1, 'Accountant');

insert into customer values(621, 'Sarvesh Kaushik', 18, 4, 'Project Manager');

insert into customer values(622, 'Dave Skylark', 22, 4, 'Anchor');

insert into customer values(623, 'Dr drew', 25, 24, 'Doctor');

insert into customer values(624, 'Maximo', 29, 11, 'Teacher');



SELECT \* FROM ORDERS;

Graphical user interface, text, application, Word

Description automatically generated

SELECT \* FROM SALES;

Graphical user interface, text, application, Word

Description automatically generated

**[ Action 1 / Operation 1]**

Customer X orders two office supply items from an online Web site.

Consider the following query, Providing the resultant data set for customer who has placed the order.

When the customer will buy two new office supply items the sales table will get updated. Two new records will be placed inside the sales table. Consider the following Orders made by customer Sarvesh Kaushik.

SELECT CUSTOMER.CUSTOMER\_ID, orders.order\_id,customer.cust\_name FROM CUSTOMER INNER JOIN ORDERS ON CUSTOMER.CUSTOMER\_ID = ORDERS.CUSTOMER\_ID WHERE customer.customer\_id =621;

Graphical user interface, text, application

Description automatically generated

Consider the following PL/SQL procedure will get executed when customer will initiate the purchase,

CREATE OR REPLACE PROCEDURE purchasingitems(Customer\_ID INT)

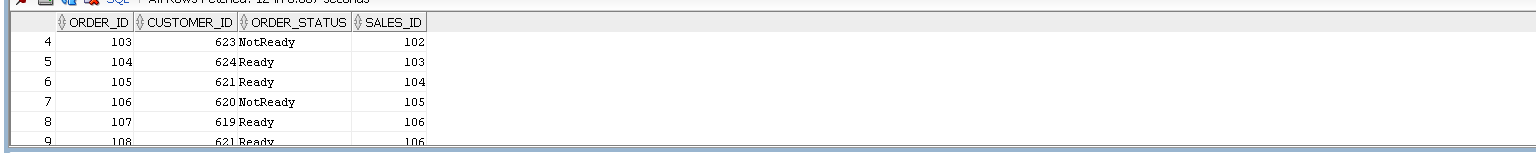
IS

BEGIN

INSERT INTO orders Values(105, Customer\_ID,'Ready',104);

INSERT INTO orders Values(108, Customer\_ID,'Ready',106);

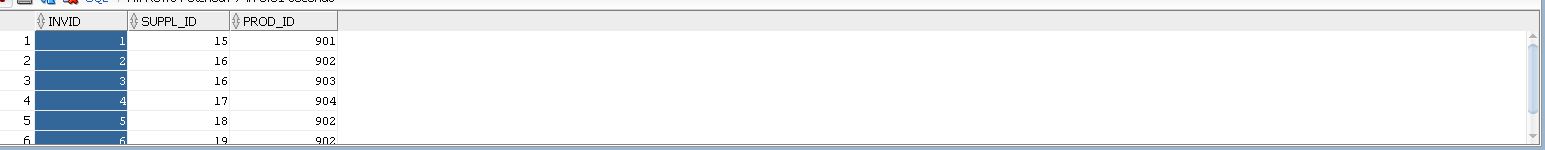
END;



**[ Action 2 / Operation 2 ]**

**The products that were ordered are verified that they exist in the supplier’s inventory.**

In order to verify if the products exist in the supplier inventory, We will create a supplier inventory table. It will help us to track the Suppliers having the product stock.



**The Warehouse is alerted to the customer’s order.**

A trigger could be used to update the details in the warehouse whenever the new record is inserted into the orders table. The trigger will make sure that Warehouse will get altered as new records are updated according to the customer’s order. Instead of writing DBMS\_OUTPUT statement Trigger will have code to update warehouse, supplied and order table.

SET SERVEROUTPUT ON;

CREATE OR REPLACE TRIGGER createalert

AFTER INSERT ON

ORDERS

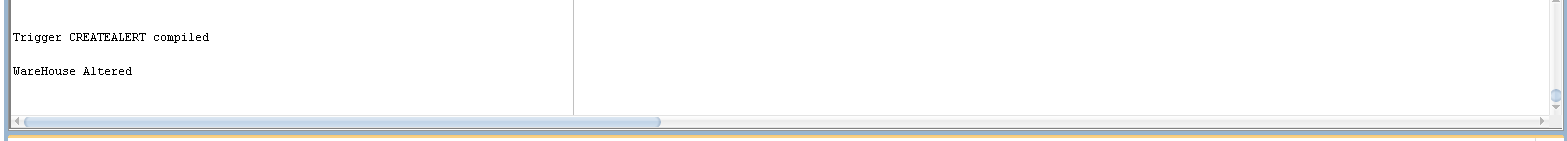
FOR EACH ROW

DECLARE ORDERID NUMBER;

BEGIN

DBMS\_OUTPUT.PUT\_LINE ('Warehouse Altered');

END;



**The customer’s product order is given to the shipping department.**

When the customers product order will be given to the shipping department the shipping id and the product id can be tracked by putting the records into the Product Shipping table that will keep record of products shipping id, shippers name and the product ID as given below.

CREATE TABLE Shippingtracker

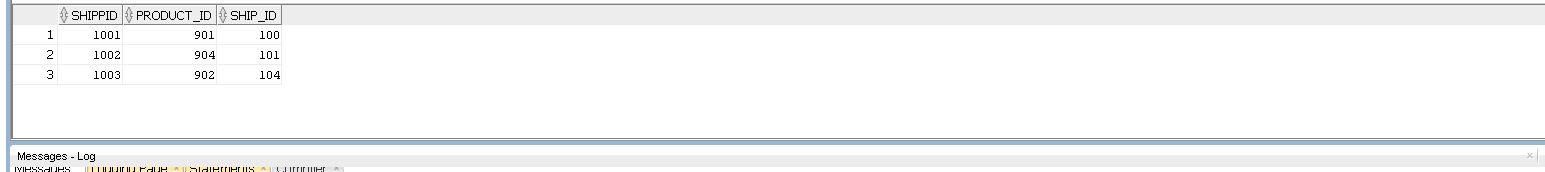
(

shippid NUMBER PRIMARY KEY,

PRODUCT\_ID INT CONSTRAINT FK\_PROD\_TEST REFERENCES PRODUCTS(PRODUCT\_ID),

SHIP\_ID NUMBER CONSTRAINT FK\_shid REFERENCES SHIPPERS(SHIPPER\_ID)

);



**The Quantity on Hand ( QOH ) is reduced by the number of items That were purchased by the customer.**

For each order placed and invoice generated the quantity of the respective product item will decrease in the table. The change can be implemented with the help of update statement query.

SELECT COUNT(\*) as OrderCount, CUSTOMER\_ID FROM ORDERS GROUP BY (CUSTOMER\_ID);

Graphical user interface, text, application

Description automatically generated

Once you get the order count then we can use the update statement.

UPDATE ORDER\_ITEM SET Quantity = Quantity – 2 where CustomerID = 621;

The Quantity will be decreased based on the count received from previous query.

**The Supplier is notified that the ordered items were removed from the seller’s inventory.**

The Supplier is notified that the ordered items were removed from the seller’s inventory.

SET SERVEROUTPUT ON;

CREATE OR REPLACE TRIGGER NotifySupplier

AFTER DELETE ON

ORDER\_ITEM

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE ('Inform Supplier');

END;

After execution of the above query it will create a trigger that will activate whenever we try to delete the data from the ORDER\_ITEM table.

**The Customer is alerted that the products have been shipped.**

The ShippingTracker Table is going to be updated when new order is shipped. A trigger created on the shipping table whenever new insert placed on Shippingtracker the customer will be alerted.

CREATE OR REPLACE TRIGGER alertShipper

AFTER INSERT ON

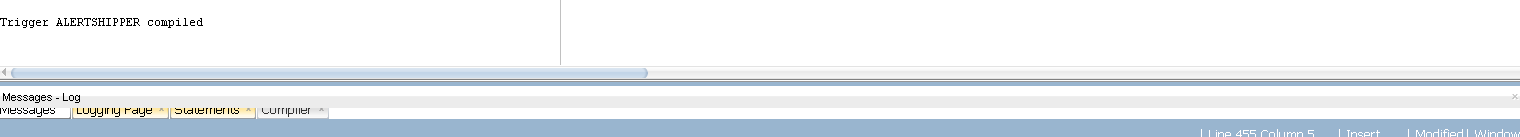
ShipingTracker

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE ('Alert Customer');

END;



**The Customer Service Department has been alerted that the customer’s order has been processed and will contact the customer when the order has been delivered.**

CREATE OR REPLACE TRIGGER Gettriggerstatus

AFTER INSERT ON

ORDERS

DECLARE

StatusVar VARCHAR(10);

BEGIN

SELECT ORDER\_STATUS INTO StatusVar FROM ORDERS WHERE ORDER\_ID = 100;

IF Statusvar = ‘Ready’ THEN

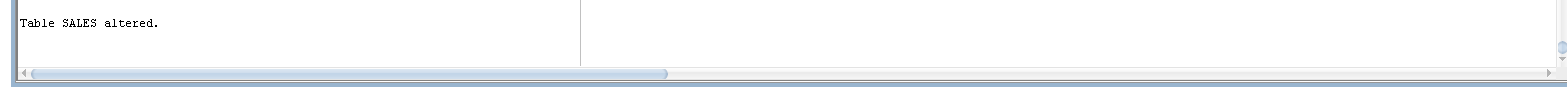
DBMS\_OUTPUT.PUT\_LINE(‘ Contact The Customers’);

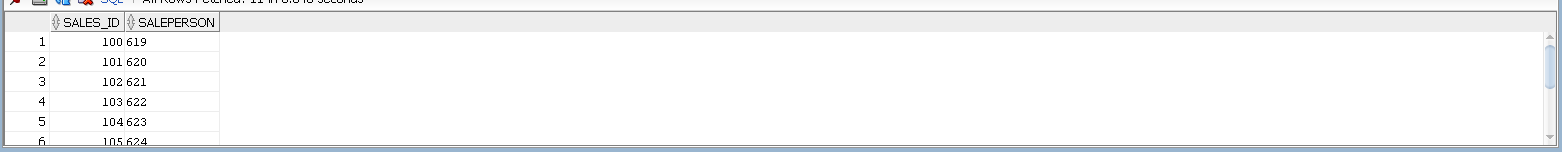
END;

**[ Action 3 / Review Your Schema ]**

**Remove order\_status from the sales table. In this manner, sales will just focus on sale\_sid and sales\_person and not on the order, which will keep the table normalized.**

ALTER TABLE SALES DROP COLUMN ORDER\_STATUS ;

****

****

**Add shipping\_Id to orders, which already has an order\_status. order\_status takes a value "Ready" or "SHIPPED or DELIVERED".**

**Ready means it is being readied for shipping, SHIPPED means in transit and DELIVERED means it was handed over to the customer.**

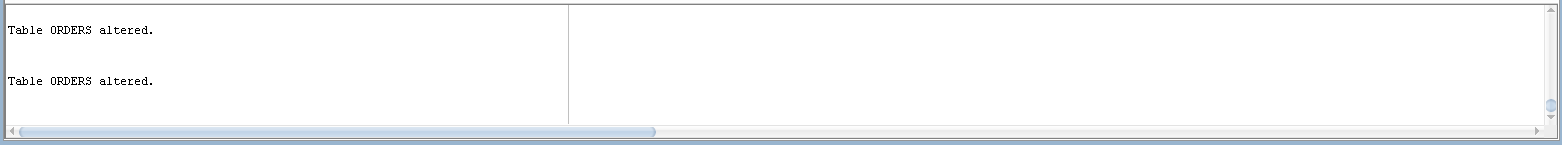
ALTER TABLE ORDERS

ADD shipping\_Id NUMBER;

ALTER TABLE ORDERS

ADD CONSTRAINT FK\_altord

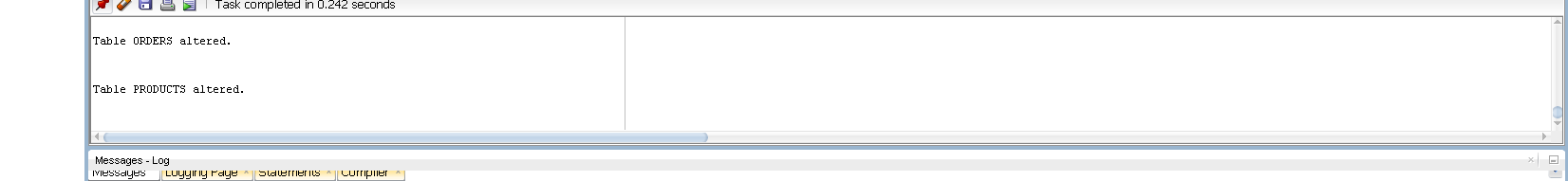
FOREIGN KEY (shipping\_Id) REFERENCES SHIPPERS(SHIPPER\_ID );



**Add quantity to the products table so that we know how much quantity the supplier has currently.**

ALTER TABLE products

ADD QUANTITY INT;



**Added a new table Warehouse\_Quantity to track the quantity available for each product at the warehouse. When Items are shipped, this table will show the reduction in the amount and new items from supplier will be added here. This tracks Quantity on Hand ( QOH ) .**

CREATE TABLE Warehouse\_Quantity(

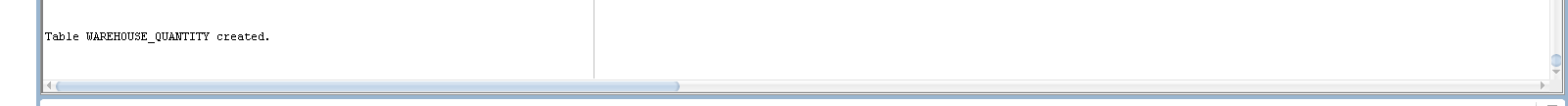
id INT PRIMARY KEY,

PRODUCT\_ID INT CONSTRAINT FK\_POTESTID REFERENCES PRODUCTS(PRODUCT\_ID),

QOH INT

);

Consider the above table will calculate the product and each respective quantity once the product order is placed.

****

Add a new table called Invoice\_Detail which tracks invoice\_id, product\_id, quantity per product and amount per product.

More information can be added here including tax, discounts, etc

CREATE TABLE INVOICE\_DETAIL

(

id INT PRIMARY KEY,

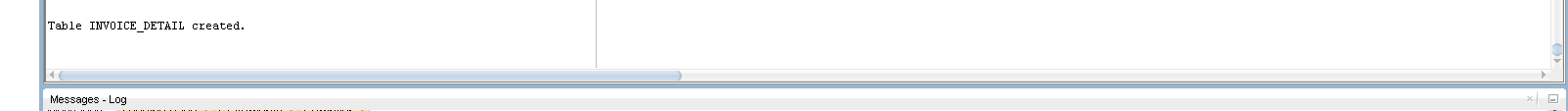
INVOICE\_ID INT CONSTRAINT FK\_INVID REFERENCES INVOICES(INVOICE\_ID),

PRODUCTT\_ID INT CONSTRAINT FK\_Inprod REFERENCES PRODUCTS(PRODUCT\_ID ),

QPC INT,

amountPP INT

);

****

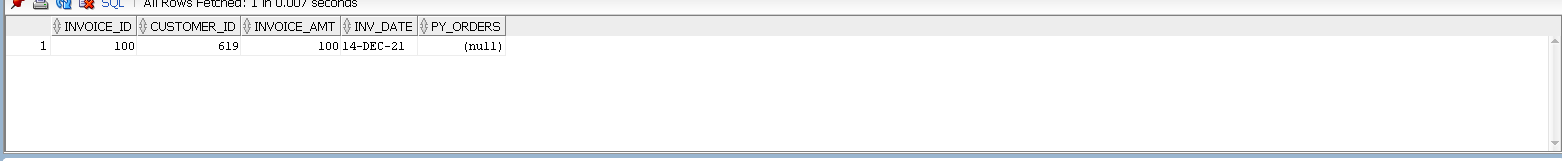
**STEP4:**

**Perform each of these analytical functions on your updated schema.**

**Shipment status / lag time (analytics) / lead times ( analytics )**

**Lag Time:** Lag is an analytical function that gives access to the rows that comes before the current row at a particular offset.

SELECT INVOICE\_ID, CUSTOMER\_ID, INVOICE\_AMT, Inv\_Date, LAG(INVOICE\_AMT) OVER (ORDER BY Inv\_Date) py\_orders FROM INVOICES WHERE INVOICE\_ID =100;

****

As the Invoice\_ID = 100 is the first record in the table the lag is returning the null value as there is no other record before it.

**Lead Time:** Lead is an analytical function that gives access to the rows that comes after the current row at a particular offset**.**

SELECT INVOICE\_ID, CUSTOMER\_ID, INVOICE\_AMT, Inv\_Date, LEAD(CUSTOMER\_ID) OVER (ORDER BY Inv\_Date) py\_orders FROM INVOICES WHERE INVOICE\_ID =105;

The 105 is the last record in the Invoices table so Lead function will return null as there is not other record after the last record in the INVOICES table.

**Text

Description automatically generated with low confidence**

SELECT \* FROM INVOICES;

**Graphical user interface, application, Word

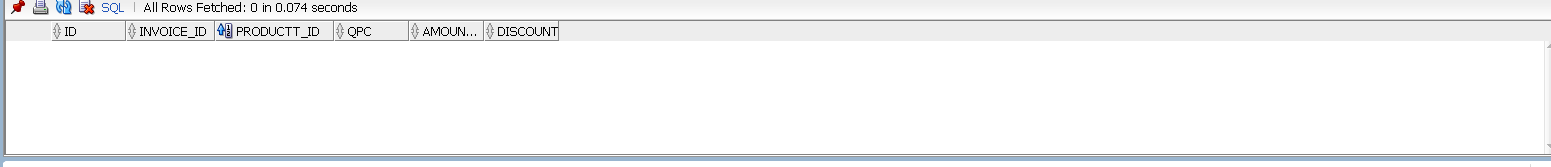
Description automatically generated**

**1>Explain how your supply chain schema would handle the promotion of a product. For example, this week only the distributor is offering a 10 for $ 10 sale on certain office supplies.**

The promotion for the supply chain schema would be handled with the help of invoice\_detail table. As the invoice\_detail table consist of Invoice Id, product id, quantity per product and amount per product. It will be having one more field as a Sales discount.

ALTER TABLE INVOICE\_DETAIL

ADD Discount INT.

****

**2> Explain how you would use your database schema to alert the distributor to replenish the warehouse.**

The Database Schema could be effectively use the triggers to alert the warehouse distributer the stock needs to be replenished.

CREATE OR REPLACE TRIGGER alertWarehouse

AFTER INSERT ON

INVOICES

FOR EACH ROW

DECLARE

quantity\_onhand INTEGER;

minimumquantity := 10;

BEGIN

SELECT QOH INTO quantity\_onhand FROM Warehouse\_Quantity where ID=2;

IF quantity\_onhand< minimumquantity THEN

DBMS\_OUTPUT.PUT\_LINE ('Alert Customer');

END IF;

END;

The above trigger will be executed whenever there is insert on the Invoices table. When the new invoice is generated user has purchased a new product and that means the product quantity in the Warehouse\_Quantity table has reduced.

**(3) If you were designing some PL / SQL triggers for this application, list at least five triggers that you would design.**

1. **The Trigger to create alert when the suppliers inventory is low: This Trigger will help to understand suppliers when the inventory is low and help them to refill the stock to fill the inventory for smooth functioning of orders.**

CREATE OR REPLACE TRIGGER alertWarehouse

AFTER INSERT ON

INVOICES

FOR EACH ROW

DECLARE

quantity\_onhand INTEGER;

minimumquantity := 10;

BEGIN

SELECT QOH INTO quantity\_onhand FROM Warehouse\_Quantity where ID=2;

IF quantity\_onhand< minimumquantity THEN

DBMS\_OUTPUT.PUT\_LINE ('Alert Customer');

END IF;

END;

1. **Alerting the Customers once the Shipper has shipped the order: Customer notification when the shipper has shipped the order. It can be implemented by creating a Trigger to alert the customers as below.**

CREATE OR REPLACE TRIGGER alertCustomer

AFTER INSERT ON

ShipingTracker

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE ('Alert Customer');

END;

above trigger can be modified by declaring the variables and checking for new Shipper ID. Whenever there is a new insert in the ShippingTracker table the mail will be sent to the corresponding customer\_id in the invoices table which is connected with the ORDER\_ITEM table having the product\_id in it.

1. Notifying the customers when their order is ready is one of the important features of the database schema.

CREATE OR REPLACE TRIGGER informorderstatus

AFTER INSERT ON ORDERS

FOR EACH ROW

DECLARE

orderstat VARCHAR;

BEGIN

SELECT ORDER\_STATUS INTO orderstat FROM ORDERS;

IF orderstat = ‘Ready’ THEN

DBMS\_OUTPUT.PUT\_LINE(“Alter Customer On Order Status”);

END IF;

END;

1. Alerting customer about certain discount offers.

CREATE OR REPLACE TRIGGER informorderstatus

AFTER INSERT ON INVOICE\_DETAIL

FOR EACH ROW

DECLARE

CHECKDIS INT;

BEGIN

SELECT DISCOUNT INTO CHECKDIS FROM INVOICE\_DETAIL;

IF orderstat > :OLD.DISCOUNT THEN

DBMS\_OUTPUT.PUT\_LINE(“Alter Customer On INCREASED PRODUCT DISCOUNT”);

END IF;

END;

1. **Indicating supplier if a particular ProductID do not exists anymore** or permanently deleted.

CREATE OR REPLACE TRIGGER productremoved

AFTER DELETE ON INVOICE\_DETAIL

FOR EACH ROW

DECLARE

DBMS\_OUTPUT.PUT\_LINE(“Alter Supplier on the removal of the product”)

END;

**(4) One of the foundations of the supply chain is operations, which includes forecasting, Just in Time (JIT), quality, inventory management and information system design. Considering a restaurant supplies franchise, comment on how this enterprise could effectively utilizes these categories of operations.**

**Forecasting:** The forecasting in the supply chain management is one of the important features that determined overall profitability, workflow productivity and the process improvement in the organization. Forecasting is process of prediction of accurate demand, value as a price and supplying capacity of a product.

The predictive analysis plays an important role in forecasting the future sales and save resources and cost on the overproduction.

In the restaurant franchise, Forecasting would help to estimate what type of resources a restaurant would require in the upcoming months over the others. It will take customer trends and food choices into account and set the budget, demand and prices based on that**.**

**Just In Time:** Just In Time compilation process is used to provide efficient delivery system. It produced just enough to satisfy the needs. One of the widely known example for Just InTime Supply Management Chain is the Toyota’s Lean Production System. Toyota Production System is based on the Just In Time, as they manufacture the quantity based on the current needs and do not overproduce in bulk.

**Quality:** The product quality established the organizations credibility in the market. When you consider a newly launched product it gets evaluated on certain parameters such as Price, Features, Quality and Durability. The quality of the product is directly related to its popularity. The implementation of effective product lifecycle would built products with high qualities that will help organizations to accumulate more revenues.

**Inventory Management:** The effective inventory management for the restaurant chain would allow to keep necessary resources stock properly filled. As the stock is properly filled it will improve the delivery process and keep overhead costs under the budget. It will allow suppliers to keep track of most consumed resources over the least consumed resources.

**Information System Design:** It is a process of defining different modules, structures, Components, and data related to this element in the System. In the Restaurant Franchise supplier business, it will be useful in designing the software modules responsible for handling large volumes of inventory data effectively. It will boost organization workflow productivity and profitability in the longer run.

**Suppose the products ordered by Customer X have embedded RFID labeling device(s). How would your database schema be used to track the position / location of the products in the chain and to assist in quicker delivery?**

RFID is known as the Radio Frequency Identification. RFID is used to capture the data encoded in the digital labels. In RFID data is analyzed with the help of radio waves as reader. Barcode is also important measure for data tracking, RFID is more useful than barcode as does not require the proper positioning relative to the scanner.

When the product is enabled with the RFID labels, it will help customer to extract the product description through the products table created in the Supply Management Chain Project.

Customers do not need to relatively position the products to scan the code. It will save the time and increase the product usability.

In the Shipping phase RFID will be useful to understand the exact product route. As the data can be programmed and once the product is scanned it will display the customers information from the customers table including the shipping address, delivery date, and mode of transportation.

RFID will be useful for warehouse conveyor belt sorting, as a scanning machine will quickly scan the products and send it to appropriate session.