ALL TABLES AND VIEWS ARE FOUND ON ANKIT BOMBWAL’S DATABASE

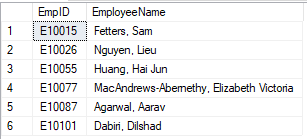
Q1)

SELECT EmpID, EmpLastName + ', ' + EmpFirstName AS EmployeeName

FROM tblEmployee

WHERE EmpID IN (SELECT DISTINCT BuyerEmpID FROM tblPurchaseOrder)

ORDER BY EmpID;



Q2)

SELECT

e.EmpID,

CONCAT(e.EmpLastName, ', ', e.EmpFirstName) AS EmployeeName,

ISNULL(manager.EmpLastName +', '+ manager.EmpFirstName, 'No Manager') AS ManagerName,

po.PONumber,

po.PODatePlaced,

v.VendorID,

v.VendorName,

UPPER(CONCAT(v.VendorCity, ', ', v.VendorState)) AS VendorLocation

FROM tblPurchaseOrder po

INNER JOIN tblEmployee e

ON po.BuyerEmpID = e.EmpID

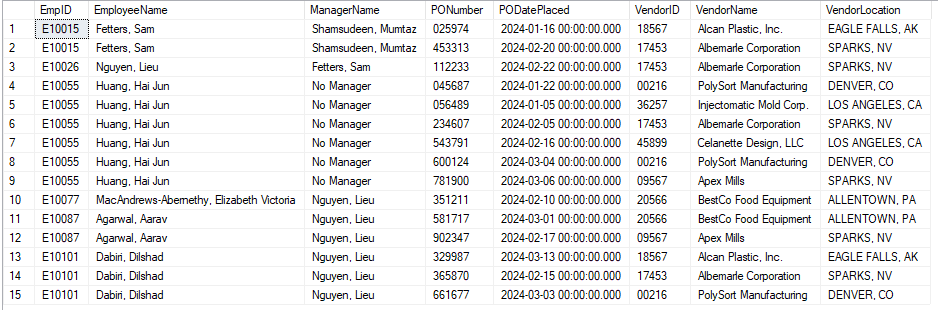
INNER JOIN tblVendor v

ON v.VendorID = po.VendorID

LEFT OUTER JOIN tblEmployee manager

ON e.EmpMgrID = manager.EmpID

ORDER BY e.EmpID, po.PONumber;



Q3)

CREATE VIEW vw\_TotalReceivedQuantity AS

SELECT POLineID, SUM(QtyReceived) AS TotalReceived

FROM tblReceiver

GROUP BY POLineID;

SELECT pol.PONumber,

vend.VendorName,

pol.ProductID,

prod.ProductDescription,

DateNeeded AS ProductDateNeeded,

QtyOrdered,

ISNULL(trq.TotalReceived,0) AS TotalQtyReceived,

QuantityRemaining = QtyOrdered - ISNULL(trq.TotalReceived, 0),

CASE

WHEN ISNULL(trq.TotalReceived, 0) = 0 THEN 'Not Received'

WHEN pol.QtyOrdered - ISNULL(trq.TotalReceived, 0) = 0 THEN 'Complete'

WHEN pol.QtyOrdered - ISNULL(trq.TotalReceived, 0) < 0 THEN 'Over Shipped'

WHEN pol.QtyOrdered - ISNULL(trq.TotalReceived, 0) > 0 THEN 'Partially Received'

END 'ReceivingStatus'

FROM tblPurchaseOrderLine pol

LEFT OUTER JOIN vw\_TotalReceivedQuantity trq

ON trq.POLineID = pol.POLineID

INNER JOIN tblProduct prod

ON prod.ProductID = pol.ProductID

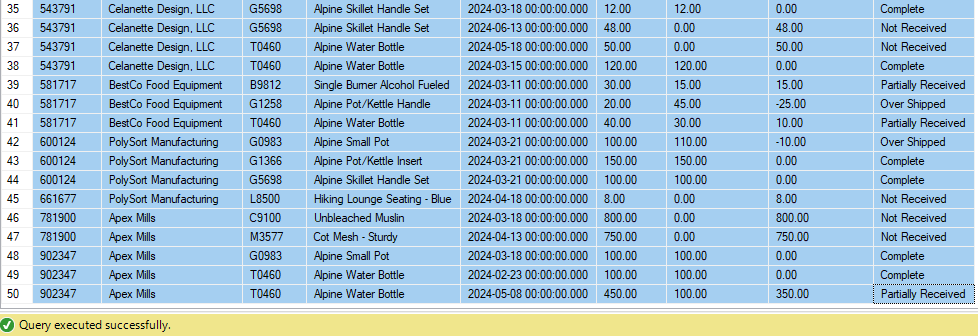
INNER JOIN tblPurchaseOrder po

ON po.PONumber = pol.PONumber

INNER JOIN tblVendor vend

ON vend.VendorID = po.VendorID

ORDER BY pol.PONumber, pol.ProductID;



Q4)

CREATE VIEW vw\_OpenOrders AS

SELECT pol.PONumber

FROM tblPurchaseOrderLine pol

LEFT OUTER JOIN vw\_TotalReceivedQuantity trq

ON trq.POLineID = pol.POLineID

INNER JOIN tblPurchaseOrder po

ON po.PONumber = pol.PONumber

WHERE pol.QtyOrdered - ISNULL(trq.TotalReceived, 0) > 0 OR ISNULL(trq.TotalReceived, 0) = 0

GROUP BY pol.PONumber

SELECT pol.PONumber,

CONVERT(VARCHAR, po.PODatePlaced, 107) AS PODatePlaced,

CONVERT(VARCHAR, po.PODateNeeded, 107) AS PODateNeeded,

vend.VendorName

FROM tblPurchaseOrderLine pol

LEFT OUTER JOIN vw\_OpenOrders oo

ON oo.PONumber = oo.PONumber

INNER JOIN tblPurchaseOrder po

ON po.PONumber = pol.PONumber

INNER JOIN tblVendor vend

ON vend.VendorID = po.VendorID

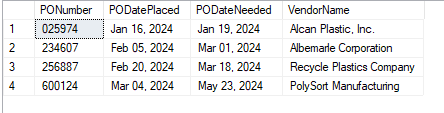
WHERE pol.PONumber NOT IN

(SELECT PONumber

FROM vw\_OpenOrders)

OR pol.PONumber IS NULL

GROUP BY pol.PONumber, po.PODatePlaced, po.PODateNeeded,vend.VendorName



Q5)

CREATE VIEW vw\_OldestQuantity AS

SELECT ph.ProductID,

ph.Qty AS OldestQty

FROM tblPurchaseHistory ph

INNER JOIN vw\_HistoryData hd

ON ph.ProductID = hd.ProductID

GROUP BY ph.ProductID, ph.Qty, ph.DatePurchased, hd.OldestPastPurchaseDate

HAVING ph.DatePurchased = hd.OldestPastPurchaseDate

CREATE VIEW vw\_HistoryData AS

SELECT ProductID,

MAX(Price) AS PastMaxPrice,

MIN(Price) AS PastMinPrice,

AVG(Price) AS PastAvgPrice,

CONVERT(VARCHAR, MIN(DatePurchased), 107) AS OldestPastPurchaseDate

FROM tblPurchaseHistory

GROUP BY ProductID;

CREATE VIEW vw\_OldestPrice AS

SELECT ph.ProductID,

ph.Price AS OldestPastPurchasePrice

FROM tblPurchaseHistory ph

INNER JOIN vw\_HistoryData hd

ON ph.ProductID = hd.ProductID

GROUP BY ph.ProductID, ph.Price, ph.DatePurchased, hd.OldestPastPurchaseDate

HAVING ph.DatePurchased = hd.OldestPastPurchaseDate

SELECT prod.ProductID,

prod.ProductDescription,

prod.EOQ AS ProductEconomicOrderQuantity,

hd.OldestPastPurchaseDate AS OldestPurchaseDate,

ISNULL(oq.OldestQty, 0) AS QuantityPurchased,

ISNULL(op.OldestPastPurchasePrice,0) AS PurchasePrice

FROM tblProduct prod

LEFT OUTER JOIN vw\_HistoryData hd

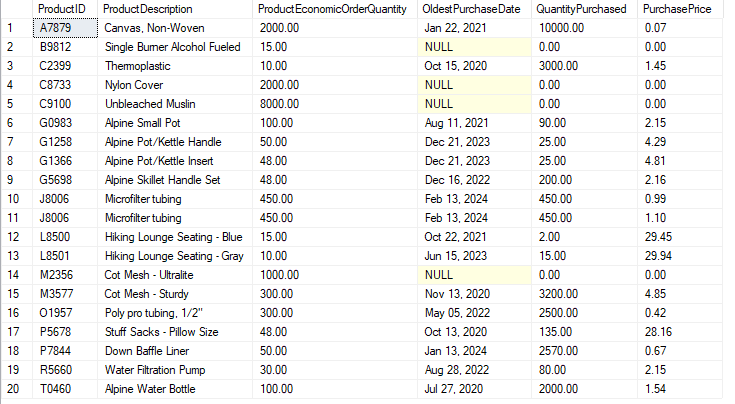
ON hd.ProductID = prod.ProductID

LEFT OUTER JOIN vw\_OldestPrice op

ON op.ProductID = prod.ProductID

LEFT OUTER JOIN vw\_OldestQuantity oq

ON oq.ProductID = prod.ProductID



Q6)

SELECT pol.ProductID,

prod.ProductDescription,

op.OldestPastPurchasePrice AS OldestHistoricalPrice,

pol.Price AS CurrentPrice,

po.PONumber,

vend.VendorName

FROM tblPurchaseOrderLine pol

INNER JOIN tblProduct prod

ON prod.ProductID = pol.ProductID

INNER JOIN tblPurchaseOrder po

ON po.PONumber = pol.PONumber

INNER JOIN tblVendor vend

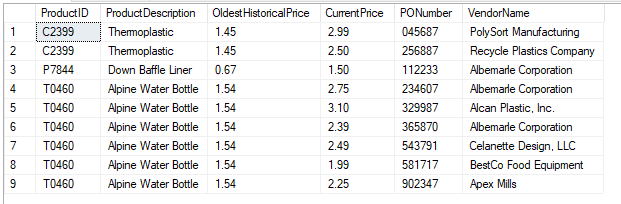
ON vend.VendorID = po.VendorID

LEFT OUTER JOIN vw\_OldestPrice op

ON op.ProductID = pol.ProductID

WHERE pol.Price > 1.25\*op.OldestPastPurchasePrice

ORDER BY pol.ProductID, po.PONumber



Q7)

SELECT

VendorID AS Vendor#,

COUNT(vw\_Overshipment.ReceivingStatus) AS POLineCount

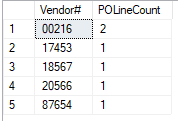
FROM vw\_Overshipment

INNER JOIN tblPurchaseOrder

ON vw\_Overshipment.PONumber = tblPurchaseOrder.POnumber

WHERE ReceivingStatus='Over Shipped'

GROUP BY VendorID



Q8)

SELECT

VendorID,

VendorName,

VendorEmail,

POLineCount AS 'CountofOver-ShippedPurchaseOrderLines'

FROM tblVendor

INNER JOIN vw\_Q7Data

ON tblVendor.VendorID=vw\_Q7Data.Vendor#

WHERE POLineCount=(SELECT MAX(POLineCount) FROM vw\_Q7Data)



Q9)

CREATE VIEW vw\_EmpReceiver AS

SELECT

worker.EmpID ,

worker.EmpLastName EmpLast,

worker.EmpFirstName EmpFirst,

worker.EmpEmail,

mgr.EMPID MgrID,

mgr.EmpLastName MgrLast,

mgr.EmpFirstName MgrFirst,

mgr.EmpEmail MgrEmail,

SUM(QtyReceived) AS TotalDamagedItemsReceived

FROM tblReceiver

INNER JOIN tblEmployee AS worker

ON tblReceiver.ReceiveEmpID=worker.EmpID

LEFT OUTER JOIN tblEmployee AS mgr

ON worker.EmpMgrID=mgr.EmpID

LEFT OUTER JOIN tblCondition

ON tblReceiver.ConditionID=tblCondition.ConditionID

WHERE ConditionDescription LIKE '%damage%'

GROUP BY worker.EmpID, worker.EmpLastName,worker.EmpFirstName, mgr.EMPID,mgr.EmpLastName,mgr.EmpFirstName,mgr.EmpEmail,worker.EmpEmail

SELECT

EmpID AS EmployeeID,

CONCAT(EmpLast,', ',SUBSTRING(EmpFirst,1,1)) AS EmployeeName,

EmpEmail,

MgrID AS ManagerEmpID,

CONCAT(MgrLast,', ',SUBSTRING(MgrFirst,1,1)) AS ManagerName,

MgrEmail AS ManagerEmail,

TotalDamagedItemsReceived

FROM vw\_EmpReceiver

WHERE TotalDamagedItemsReceived=(SELECT MAX(TotalDamagedItemsReceived) FROM vw\_EmpReceiver)



Q10)

CREATE VIEW vw\_CurrentData AS

SELECT ProductID,

SUM(QtyOrdered) AS CurrentQtyOnOrder,

MAX(Price) AS CurrentMaxPrice,

MIN(Price) AS CurrentMinPrice,

AVG(Price) AS CurrentAvgPrice

FROM tblPurchaseOrderLine

GROUP BY ProductID;

CREATE VIEW vw\_HistoryData AS

SELECT ProductID,

MAX(Price) AS PastMaxPrice,

MIN(Price) AS PastMinPrice,

AVG(Price) AS PastAvgPrice,

CONVERT(VARCHAR, MIN(DatePurchased), 107) AS OldestPastPurchaseDate

FROM tblPurchaseHistory

GROUP BY ProductID;

CREATE VIEW vw\_OldestPrice AS

SELECT ph.ProductID,

ph.Price AS OldestPastPurchasePrice

FROM tblPurchaseHistory ph

INNER JOIN vw\_HistoryData hd

ON ph.ProductID = hd.ProductID

GROUP BY ph.ProductID, ph.Price, ph.DatePurchased, hd.OldestPastPurchaseDate

HAVING ph.DatePurchased = hd.OldestPastPurchaseDate

SELECT prod.ProductID,

prod.ProductDescription,

ISNULL(cd.CurrentQtyOnOrder, 0) AS CurrentQtyOnOrder,

ISNULL(cd.CurrentMaxPrice, 0) AS CurrentMaxPrice,

ISNULL(cd.CurrentMinPrice, 0) AS CurrentMinPrice,

ISNULL(cd.CurrentAvgPrice, 0) AS CurrentAvgPrice,

ISNULL(hd.PastMaxPrice, 0) AS PastMaxPrice,

ISNULL(hd.PastMinPrice, 0) AS PastMinPrice,

ISNULL(hd.PastAvgPrice, 0) AS PastAvgPrice,

ISNULL(hd.OldestPastPurchaseDate, 'No Previous Purchase') AS OldestPastPurchaseDate,

ISNULL(op.OldestPastPurchasePrice, 0) AS OldestPastPurchasePrice

FROM tblProduct prod

LEFT OUTER JOIN vw\_CurrentData cd

ON prod.ProductID = cd.ProductID

LEFT OUTER JOIN vw\_HistoryData hd

ON prod.ProductID = hd.ProductID

LEFT OUTER JOIN vw\_OldestPrice op

ON prod.ProductID = op.ProductID

ORDER BY prod.ProductID, -op.OldestPastPurchasePrice

