Caleb Hayden

CIS 310-01

Assignment 8

Due 11/1/17

--1

SELECT ITEMID, DESCRIPTION, LISTPRICE

FROM PET..MERCHANDISE

HAVING LISTPRICE> AVG(LISTPRICE)

WHERE LISTPRICE > (SELECT AVG(LISTPRICE) FROM PET..MERCHANDISE)

--2

CREATE VIEW Averages AS

SELECT

O.ItemID,

AVG(O.Cost) AS AverageCost,

AVG(S.SalePrice) AS AveragePrice

FROM

PET..OrderItem O INNER JOIN PET..SaleItem S ON O.ItemID = S.ItemID

GROUP BY

O.ItemID

SELECT

ItemID,

AverageCost,

AveragePrice

FROM

Averages A

WHERE

AveragePrice > (SELECT 1.5 \* AverageCost

FROM Averages B

WHERE A.ItemID = B.ItemID)

--3

CREATE VIEW TOTALSALES AS

SELECT E.EMPLOYEEID, E.LASTNAME, SUM(I.SALEPRICE \* I.Quantity) AS [TOTALSALES]

FROM PET..EMPLOYEE E INNER JOIN PET..SALE S ON E.EMPLOYEEID = S.EMPLOYEEID INNER JOIN

PET..SALEITEM I ON S.SALEID = I.SALEID

GROUP BY E.EMPLOYEEID, E.LASTNAME

HAVING SUM(I.SALEPRICE \* I.Quantity) <> 0

SELECT T.EMPLOYEEID, T.LASTNAME, T.TOTALSALES, T.TOTALSALES / (SELECT SUM(TOTALSALES) FROM TOTALSALES) AS [PCTSALES]

FROM TOTALSALES T INNER JOIN PET..SALE S ON T.EMPLOYEEID = S.EMPLOYEEID INNER JOIN

PET..SALEITEM I ON S.SA90-=LEID = I.SALEID

GROUP BY T.EMPLOYEEID, T.LASTNAME, T.TOTALSALES

--4

CREATE VIEW ShippingPct AS

SELECT S.SUPPLIERID, S.NAME, SUM(M.SHIPPINGCOST) / SUM((O.COST\*O.QUANTITY)+M.SHIPPINGCOST) AS [PCTSHIPCOST]

FROM PET..SUPPLIER S INNER JOIN PET..MERCHANDISEORDER M ON S.SUPPLIERID = M.SUPPLIERID INNER JOIN

PET..ORDERITEM O ON M.PONUMBER = O.PONUMBER

GROUP BY S.SUPPLIERID, S.NAME

SELECT SupplierID, NAME, PCTSHIPCOST

FROM ShippingPct

WHERE PCTSHIPCOST = (SELECT MAX(PCTSHIPCOST)

FROM ShippingPct)

--5

CREATE VIEW CustomerTotal AS

SELECT C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME, SUM(I.SALEPRICE) AS [MERCTOTAL], SUM(A.SALEPRICE) AS [ANIMALTOTAL], SUM(I.SALEPRICE) + SUM(A.SALEPRICE) AS [GRANDTOTAL]

FROM PET..SALEANIMAL A INNER JOIN PET..SALE S ON A.SALEID = S.SALEID INNER JOIN

PET..SALEITEM I ON S.SALEID = I.SALEID INNER JOIN

PET..CUSTOMER C ON S.CUSTOMERID = C.CUSTOMERID

GROUP BY C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME

SELECT \*

FROM CustomerTotal

WHERE ANIMALTOTAL = (SELECT MAX(ANIMALTOTAL) FROM CustomerTotal)

--6

CREATE VIEW MAYSALES AS

SELECT C.CUSTOMERID, SUM(I.SALEPRICE\*I.Quantity) AS [MAYTOTAL], S.SaleDate

FROM PET..CUSTOMER C INNER JOIN PET..SALE S ON C.CUSTOMERID = S.CUSTOMERID INNER JOIN

PET..SALEITEM I ON S.SALEID = I.SALEID

--WHERE S.SALEDATE LIKE '%MAY%'

GROUP BY C.CUSTOMERID, S.SaleDate

ORDER BY S.SaleDate

SELECT M.CUSTOMERID, C.LASTNAME, C.FIRSTNAME, M.MAYTOTAL

FROM PET..CUSTOMER C INNER JOIN MAYSALES M ON C.CUSTOMERID = M.CUSTOMERID

WHERE M.MAYTOTAL > 100

AND M.CUSTOMERID IN

(SELECT DISTINCT C.CUSTOMERID

FROM PET..SALEITEM I INNER JOIN PET..SALE S ON I.SALEID = S.SALEID INNER JOIN

PET..CUSTOMER C ON S.CUSTOMERID = C.CUSTOMERID

WHERE S.SALEDATE LIKE '%OCT%'

GROUP BY C.CUSTOMERID

HAVING SUM(I.SALEPRICE \* I.Quantity) > 50

)

GROUP BY M.CUSTOMERID, C.LASTNAME, C.FIRSTNAME, M.MAYTOTAL

--7

CREATE VIEW PURCHASED AS

SELECT SUM(O.QUANTITY) AS SumQuantity, O.ITEMID

FROM PET..ORDERITEM O INNER JOIN PET..MERCHANDISEORDER M ON O.PONUMBER = M.PONUMBER

WHERE O.ITEMID = '16' AND

(M.ORDERDATE BETWEEN '2004-01-01' AND '2004-07-01')

GROUP BY ItemID

CREATE VIEW SOLD AS

SELECT SUM(I.QUANTITY) AS SumQuantity, I.ITEMID

FROM PET..SALEITEM I INNER JOIN PET..SALE S ON I.SALEID = S.SALEID

WHERE I.ITEMID = '16' AND

(S.SALEDATE BETWEEN '2004-01-01' AND '2004-07-01')

GROUP BY ItemID

SELECT M.DESCRIPTION, M.ITEMID, P.SumQuantity AS [PURCHASED], S.SumQuantity AS [SOLD],

(P.SumQuantity - S.SumQuantity) AS [NETINCREASE]

FROM PET..MERCHANDISE M INNER JOIN PURCHASED P ON M.ITEMID = P.ITEMID INNER JOIN

SOLD S ON M.ITEMID = S.ITEMID

--8

SELECT DISTINCT M.ITEMID, M.DESCRIPTION, M.LISTPRICE

FROM PET..MERCHANDISE M INNER JOIN PET..SALEITEM I ON M.ITEMID = I.ITEMID INNER JOIN

PET..SALE S ON I.SALEID = S.SALEID

WHERE M.LISTPRICE > 50 AND

M.ITEMID NOT IN

(SELECT I.ITEMID

FROM PET..SALEITEM I INNER JOIN PET..SALE S ON I.SALEID = S.SALEID

WHERE S.SALEDATE BETWEEN '2004-07-01' AND '2004-07-31' AND

M.ItemID = I.ItemID)

--9

SELECT DISTINCT M.ITEMID, M.DESCRIPTION, M.QUANTITYONHAND

FROM PET..MERCHANDISE M LEFT OUTER JOIN PET..ORDERITEM O ON M.ITEMID = O.ITEMID LEFT OUTER JOIN

PET..MERCHANDISEORDER MO ON O.PONUMBER = MO.PONUMBER

WHERE M.QUANTITYONHAND > 100 AND

MO.ORDERDATE IS NULL

--10

SELECT ITEMID, DESCRIPTION, QUANTITYONHAND

FROM PET..MERCHANDISE M

WHERE M.QuantityOnHand > 100 AND

ITEMID NOT IN

(SELECT O.ITEMID

FROM PET..ORDERITEM O INNER JOIN PET..MERCHANDISEORDER MO ON O.PONUMBER = MO.PONUMBER

WHERE MO.ORDERDATE LIKE '%2004%')

--11

CREATE TABLE CATEGORY

(

CATEGORY CHAR(4),

LOW INT,

HIGH INT

)

INSERT INTO CATEGORY VALUES ('WEAK',0,200)

INSERT INTO CATEGORY VALUES ('GOOD',200, 800)

INSERT INTO CATEGORY VALUES ('BEST',800,10000)

CREATE VIEW CustomerSales AS

SELECT

C.CustomerID, C.LastName, C.FirstName, (SUM(SI.SALEPRICE) + SUM(SA.SALEPRICE)) AS GrandTotal

FROM

PET..CUSTOMER C INNER JOIN PET..SALE S ON C.CUSTOMERID = S.CUSTOMERID INNER JOIN

PET..SALEITEM SI ON S.SALEID = SI.SALEID INNER JOIN

PET..SALEANIMAL SA ON S.SALEID = SA.SALEID

GROUP BY

C.CustomerID, C.LastName, C.FirstName

SELECT CUSTOMERID, LASTNAME, FIRSTNAME, GRANDTOTAL, CATEGORY.CATEGORY

FROM CATEGORY INNER JOIN CustomerSales ON CustomerSales.GRANDTOTAL < CATEGORY.HIGH AND

CustomerSales.GRANDTOTAL >= CATEGORY.LOW

--12

SELECT DISTINCT S.NAME,'Merchandise' AS 'Type of Sale'

FROM PET..SUPPLIER S INNER JOIN PET..MERCHANDISEORDER MO ON S.SUPPLIERID = MO.SUPPLIERID

WHERE MONTH(MO.ORDERDATE) = 6

UNION

SELECT DISTINCT S.NAME,' Animal' AS 'Type of Sale'

FROM PET..SUPPLIER S INNER JOIN PET..ANIMALORDER AO ON S.SUPPLIERID = AO.SUPPLIERID

--13

WHERE MONTH(AO.ORDERDATE) = 6

DROP TABLE CATEGORY

CREATE TABLE CATEGORY

(

CATEGORY CHAR(4),

LOW INT,

HIGH INT

)

--14

INSERT INTO CATEGORY VALUES ('WEAK',0,200)

SELECT \*

FROM Category

--15

UPDATE Category

SET High = 400

WHERE Category = 'Weak'

--17

DELETE FROM Category

WHERE Category = 'Weak'

--18

SELECT \*

INTO EmployeeCopy

FROM PET..Employee

DELETE FROM EmployeeCopy