

Patrick Warren

Assignment 8 – A8



November 1, 2017

CIS310-01

--Patrick Warren

--Assignment 8

--CIS 310-01

--Due: 11/01/2017

--1

SELECT ITEMID, Description, ListPrice

FROM PET..MERCHANDISE

WHERE LISTPRICE > ALL (SELECT AVG(LISTPRICE)

FROM PET..MERCHANDISE)

--2

SELECT S.ITEMID, AVG((S.SalePRICE\*S.QUANTITY)/S.QUANTITY) AS AVERAGE\_SALE\_COST, AVG((O.COST\*O.Quantity)/O.QUANTITY) AS AVERAGE\_COST

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

GROUP BY S.ItemID

HAVING AVG((S.SalePRICE\*S.QUANTITY)/S.QUANTITY) > (AVG((O.COST\*O.Quantity)/O.QUANTITY) \* 1.5)

ORDER BY S.ITEMID

--3. List the employees and their total merchandise sales expressed as a percentage of total merchandise sales for all employees.

SELECT EMPLOYEEID, SUM(SI.SALEPRICE) AS PERC\_SALES

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

GROUP BY EMPLOYEEID

HAVING (SUM(SI.SALEPRICE)/(SELECT SUM(SALEPRICE) FROM PET..SALEITEM))\*100 > 0

--4 On average, which supplier charges the highest shipping cost as a percent of the merchandise order total?

--SupplierID Name PctShipCost

SELECT S.SUPPLIERID, S.NAME, (AVG(MO.ShippingCosT)/SUM(O.QUANTITY\*O.COST))\*100 AS PCTSHIPCOST

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

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

GROUP BY S.SUPPLIERID, S.NAME

HAVING (AVG(MO.ShippingCosT)/SUM(O.QUANTITY\*O.COST))\*100 = (SELECT TOP 1 (AVG(MO.ShippingCosT)/SUM(O.QUANTITY\*O.COST))\*100 AS PCTSHIPCOST

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

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

GROUP BY S.SUPPLIERID, S.NAME

ORDER BY PCTSHIPCOST DESC

)

--5 Which customer has given us the most total money for animals and merchandise?

--CustomerID LastName FirstName Total Total GrandTotal

CREATE VIEW TOTALANIMALSALES AS

SELECT C.CUSTOMERID, SUM(SA.SALEPRICE) AS TOTAL\_ANIMAL\_SALE

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

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

GROUP BY C.CUSTOMERID

ORDER BY C.CUSTOMERID

CREATE VIEW TOTALITEMSALES AS

SELECT C.CUSTOMERID, SUM(SI.SALEPRICE\*SI.QUANTITY) AS TOTAL\_ITEM\_SALE

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

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

GROUP BY C.CUSTOMERID

ORDER BY C.CUSTOMERID

SELECT C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME, SUM(TAS.TOTAL\_ANIMAL\_SALE + TIS.TOTAL\_ITEM\_SALE) AS GRANDTOTAL

FROM TOTALANIMALSALES TAS INNER JOIN TOTALITEMSALES TIS ON TAS.CUSTOMERID = TIS.CUSTOMERID

INNER JOIN PET..CUSTOMER C ON TIS.CUSTOMERID = C.CUSTOMERID

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

HAVING SUM(TAS.TOTAL\_ANIMAL\_SALE + TIS.TOTAL\_ITEM\_SALE) = (SELECT TOP 1 SUM(TAS.TOTAL\_ANIMAL\_SALE + TIS.TOTAL\_ITEM\_SALE) AS GRANDTOTAL

FROM TOTALANIMALSALES TAS INNER JOIN TOTALITEMSALES TIS ON TAS.CUSTOMERID = TIS.CUSTOMERID

INNER JOIN PET..CUSTOMER C ON TIS.CUSTOMERID = C.CUSTOMERID

GROUP BY C.CUSTOMERID

ORDER BY GRANDTOTAL DESC

)

ORDER BY GRANDTOTAL DESC

--6 Which customers who bought more than $100 in merchandise in May also spent more than $50 on merchandise in October?

-- CustomerID LastName FirstName MayTotal

SELECT C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME, SUM(SI.QUANTITY\*SI.SALEPRICE) AS MAY\_TOTAL

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

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

WHERE MONTH(S.SALEDATE) = 5 AND C.CUSTOMERID IN (SELECT C.CUSTOMERID

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

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

WHERE MONTH(S.SALEDATE) = 10

GROUP BY C.CUSTOMERID

HAVING SUM(SI.QUANTITY\*SI.SALEPRICE) > 50

)

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

HAVING SUM(SI.QUANTITY\*SI.SALEPRICE) > 100

--7 What was the net change in quantity on hand for premium canned dog food between January 1 and July 1?

--Description ItemID Purchased Sold NetIncrease

SELECT M.DESCRIPTION, M.ITEMID, (SELECT SUM(SI.QUANTITY)

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

WHERE ITEMID IN (SELECT ITEMID

FROM PET..MERCHANDISE

WHERE DESCRIPTION = 'DOG FOOD-CAN-PREMIUM')

AND S.SALEDATE >= '2004-1-1'

AND S.SALEDATE <= '2004-7-1') AS SOLD, SUM(OI.QUANTITY) - (SELECT SUM(SI.QUANTITY)

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

WHERE SI.ITEMID IN (SELECT ITEMID

FROM PET..MERCHANDISE

WHERE DESCRIPTION = 'DOG FOOD-CAN-PREMIUM')

AND S.SALEDATE >= '2004-1-1'

AND S.SALEDATE <= '2004-7-1') AS NET\_INCREASE

FROM PET..MERCHANDISE M INNER JOIN PET..ORDERITEM OI ON OI.ITEMID = M.ITEMID

INNER JOIN PET..MERCHANDISEORDER MO ON MO.PONUMBER = OI.PONUMBER

WHERE DESCRIPTION = 'DOG FOOD-CAN-PREMIUM' AND RECEIVEDATE >= '2004-1-1' AND RECEIVEDATE <= '2004-7-1'

GROUP BY M.DESCRIPTION, M.ITEMID

--8 Which are the merchandise items with a list price of more than $50 and no sales in July?

-- ItemID Description ListPrice

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

FROM PET..MERCHANDISE M INNER JOIN PET..SALEITEM SI ON M.ItemID = SI.ItemID

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

WHERE M.LISTPRICE > 50 AND MONTH(S.SALEDATE) <> 7

GROUP BY M.ITEMID, M.DESCRIPTION, M.LISTPRICE

--9 Which merchandise items with more than 100 units on hand have not been ordered in 2004? Use an outer join to answer the question.

-- ItemID Description QuantityOnHand ItemID

SELECT M.ITEMID, M.DESCRIPTION, M.QUANTITYONHAND, OI.ITEMID, MO.ORDERDATE

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

LEFT OUTER JOIN PET..MERCHANDISEORDER MO ON MO.PONUMBER = OI.PONUMBER

WHERE M.QUANTITYONHAND > 100 AND MO.ORDERDATE IS NULL

--10 Which merchandise items with more than 100 units on hand have not been ordered in 2004? Use a subquery to answer the question.

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

FROM PET..MERCHANDISE M

WHERE M.QUANTITYONHAND > 100 AND M.ITEMID NOT IN (SELECT OI.ITEMID

FROM PET..ORDERITEM OI LEFT OUTER JOIN PET..MERCHANDISEORDER MO ON MO.PONUMBER = OI.PONUMBER

WHERE YEAR(MO.ORDERDATE) = 2004 OR MO.ORDERDATE IS NOT NULL

)

--11 Save a query to answer Exercise 5: total amount of money spent by each customer.

--Create the table shown to categorize customers based on sales. Write a query that lists

--each customer from the first query and displays the proper label.

CREATE TABLE CATEGORY

(CATEGORY CHAR(4) NOT NULL UNIQUE,

LOW SMALLINT,

HIGH SMALLINT)

ALTER TABLE CATEGORY

ADD PRIMARY KEY (CATEGORY)

INSERT INTO CATEGORY

VALUES ('BEST', 800, 10000), ('GOOD', 200, 800), ('WEAK', 0, 200)

SELECT \*

FROM CATEGORY

ORDER BY LOW

CREATE VIEW TotalCustomerSales AS

SELECT C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME, (TAS.TOTAL\_ANIMAL\_SALE) + (TIS.TOTAL\_ITEM\_SALE) AS [GRANDTOTAL]

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

INNER JOIN TOTALANIMALSALES TAS ON TAS.CUSTOMERID = S.CUSTOMERID

INNER JOIN TOTALITEMSALES TIS ON TIS.CUSTOMERID = S.CUSTOMERID

GROUP BY C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME, (TAS.TOTAL\_ANIMAL\_SALE) + (TIS.TOTAL\_ITEM\_SALE)

SELECT CUSTOMERID, LASTNAME, FIRSTNAME, GRANDTOTAL,

IIF (GRANDTOTAL <= 200, (SELECT CATEGORY FROM CATEGORY WHERE HIGH = 200),

IIF (GRANDTOTAL <= 800, (SELECT CATEGORY FROM CATEGORY WHERE HIGH = 800),

(SELECT CATEGORY FROM CATEGORY WHERE HIGH = 10000))) AS CATEGORY

FROM TotalCustomerSales

ORDER BY CUSTOMERID

--12 List all suppliers (animals and merchandise) who sold us items in June. Identify whether they sold use animals or merchandise.

--Name OrderType

SELECT SR.SUPPLIERID, NAME, 'ANIMAL' AS [ORDER TYPE], AO.ORDERDATE

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

WHERE MONTH(AO.ORDERDATE) = 6

UNION ALL

SELECT SR.SUPPLIERID, NAME, 'MERCHANDISE' AS [ORDER TYPE], MO.ORDERDATE

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

WHERE MONTH(MO.ORDERDATE) = 6

--13 Drop the table Category. Write a query to create the table Category shown in Exercise 11

DROP TABLE CATEGORY

CREATE TABLE CATEGORY

(CATEGORY CHAR(4) NOT NULL UNIQUE,

LOW SMALLINT,

HIGH SMALLINT)

ALTER TABLE CATEGORY

ADD PRIMARY KEY (CATEGORY)

SELECT \*

FROM CATEGORY

ORDER BY LOW

--14 Write a query to insert the first row of data for the table in Exercise 11.

INSERT INTO CATEGORY

VALUES ('WEAK', 0, 200)

--15 Write a query to change the High value to 400 in the first row of the table in Exercise 11.

UPDATE CATEGORY

SET HIGH = 400

WHERE CATEGORY = 'WEAK'

--17 (16?) Create a query to delete the first row of the table in Exercise 11

DELETE

FROM CATEGORY

WHERE CATEGORY = 'WEAK'

--18 (17?) Create a copy of the Employee table structure. Use a delete query to remove all data from the copy.

--Write a query to copy from the original employee table into the new one.

SELECT \* INTO EMPLOYEECOPY

FROM PET..EMPLOYEE

DELETE

FROM EMPLOYEECOPY

INSERT INTO EMPLOYEECOPY

SELECT \*

FROM PET..EMPLOYEE

SELECT \*

FROM EMPLOYEECOPY