Trisia Baltazar

Due Nov 1

Assignment 9

CIS 310-02-4178

--1

--List products with a list price higher than the average price of all products

SELECT ITEMID, DESCRIPTION, LISTPRICE

FROM PET..Merchandise

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

--2 HELP

--WHICH MERCHANDISE ITEMS HAVE AN AVERAGE SALE PRICE MORE THAN 50% HIGHER THAN THEIR AVERGAGE PURCHASE COST

SELECT A.ITEMID, AverageCost, AverageSalePrice

FROM (SELECT ITEMID, AVG(SALEPRICE) AS AverageSalePrice FROM PET..SaleItem

GROUP BY ITEMID) A INNER JOIN

(SELECT ITEMID, AVG(COST) AS AverageCost FROM PET..ORDERITEM

GROUP BY ITEMID) B ON A.ITEMID=B.ITEMID

WHERE A.AverageSalePrice>1.5\*B.AverageCost

--3

--LIST EMPLOYEES AND THEIR TOTAL MERCHANDISE SALES EXPRESSED AS A PERCENTAGE OF TOTAL MERCHANDISE SALES FOR ALL EMPLOYEES

--GET EMPLOYEE TOTAL SALES EACH

CREATE VIEW EMP\_SALES AS

SELECT E.EMPLOYEEID, SUM(SI.QUANTITY \* SI. SALEPRICE) AS EMPTOTALSALE

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

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

GROUP BY E.EmployeeID

--GET TOTAL SALES: 8541.79

CREATE VIEW TOTAL\_SALES AS

SELECT SUM(QUANTITY\*SALEPRICE) AS TOTALSALES

FROM PET..SALEITEM

--PRODUCES TABLE

SELECT E.EmployeeID, (EMP\_SALES.EMPTOTALSALE/TOTAL\_SALES.TOTALSALES) \* 100 AS PercentSale

FROM TOTAL\_SALES, EMP\_SALES INNER JOIN PET..EMPLOYEE E ON EMP\_SALES.EmployeeID = E.EmployeeID

GROUP BY E.EmployeeID, EMPTOTALSALE, TOTALSALES

--4

--On average, which supplier charges the highest shipping cost as

-- a percent of the merchandise order total?

--Get Merchandise order total per supplier

CREATE VIEW SUPPLIERTOTAL AS

SELECT MO.SUPPLIERID, SUM(QUANTITY\* COST) AS SUPPLIERMERCHTOTAL

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

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

GROUP BY MO.SUPPLIERID

--Total shipping cost

CREATE VIEW MERCH\_TOTAL AS

SELECT SUM(QUANTITY\*COST) AS TOTAL

FROM PET..ORDERITEM

SELECT AVG(SUPPLIERTOTAL.SUPPLIERMERCHTOTAL)

FROM SUPPLIERTOTAL

--SHOWS PO WITH SUPID AND TOTAL

SELECT OI.PONUMBER, MO.SUPPLIERID , SUM(QUANTITY\*COST) AS MERCHTOTAL

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

PET..ORDERITEM OI ON MO.PONUMBER = OI.PONUMBER

GROUP BY OI.PONUMBER, MO.SUPPLIERID

--5

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

--TOTAL MERCH SALES PER CUSTOMER

CREATE VIEW CUS\_MERCH\_TOTAL AS

SELECT C.CUSTOMERID, SUM(SI.QUANTITY \* SI.SALEPRICE) AS MERCHTOTAL

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

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

GROUP BY C.CUSTOMERID

--TOTAL ANIMAL SALES PER CUSTOMER

CREATE VIEW CUS\_ANIMAL\_TOTAL AS

SELECT C.CUSTOMERID, SUM(SA.SALEPRICE) AS ANIMALTOTAL

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

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

GROUP BY C.CUSTOMERID

--SUM OF BOTH TABLES

SELECT C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME,

SUM(CA.ANIMALTOTAL + CM.MERCHTOTAL) AS GRANDTOTAL

FROM PET..CUSTOMER C INNER JOIN CUS\_MERCH\_TOTAL CM ON C.CUSTOMERID = CM.CUSTOMERID INNER JOIN

CUS\_ANIMAL\_TOTAL CA ON CM.CUSTOMERID = CA.CUSTOMERID

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

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

--CUSTOMERS WHO BOUGHT MORE THAN $100 ON MERCH IN MAY

CREATE VIEW MAY100 AS

SELECT C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME, S.SALEDATE, SUM(SI.QUANTITY \*SI.SALEPRICE) AS TOTAL

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

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

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

HAVING MONTH(S.SALEDATE) = 5 AND (SUM(SI.QUANTITY \* SI.SALEPRICE) > 100)

--CUSTOMERS WHO BOUGHT MORE THAN $50 ON MERCH IN OCT

CREATE VIEW OCT50 AS

SELECT C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME, S.SALEDATE, SUM(SI.QUANTITY \*SI.SALEPRICE) AS TOTAL

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

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

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

HAVING MONTH(S.SALEDATE) = 10 AND (SUM(SI.QUANTITY \* SI.SALEPRICE) > 50)

--JOINS BOTH TABLE TO FIND CUSTOMER WITH BOTH CONDITIONS

SELECT M.CUSTOMERID, M.LASTNAME, M.FIRSTNAME, M.TOTAL AS MAYTOTAL, O.TOTAL AS OCTTOTAL

FROM MAY100 M INNER JOIN OCT50 O ON M.CUSTOMERID= O.CUSTOMERID

--7

--WHAT WAS THE NET CHANGE IN QUANTITY ON HAND FOR PREMIUM CANNED DOG FOOD

--BETWEEN JAN 1 AND JULY 1

--GET PREMIUM DOG ORDERED

CREATE VIEW TOTAL\_DOG\_FOOD\_ORDERED AS

SELECT M.ITEMID, M.DESCRIPTION, SUM(OI.QUANTITY) AS TOTALORDERED

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

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

WHERE M.DESCRIPTION LIKE'%PREMIUM%'

AND M.DESCRIPTION LIKE '%DOG%'

AND MONTH(MO.RECEIVEDATE) >=1

AND MONTH(MO.RECEIVEDATE) <7

GROUP BY M.ITEMID, M.DESCRIPTION

--GET PREMIUM DOG FOOD SOLD

CREATE VIEW TOTAL\_DOG\_FOOD\_SOLD AS

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

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

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

WHERE M.DESCRIPTION LIKE '%PREMIUM%'

AND M.DESCRIPTION LIKE '%DOG%'

AND MONTH(S.SALEDATE) >=1

AND MONTH(S.SALEDATE) <7

GROUP BY M.ITEMID, M.DESCRIPTION

--GET NET CHANGE: 336

SELECT TS.ITEMID, TS.DESCRIPTION, TORDER.TOTALORDERED - TS.TOTALSOLD AS NETCHANGEINQUANTITY

FROM TOTAL\_DOG\_FOOD\_ORDERED TORDER INNER JOIN TOTAL\_DOG\_FOOD\_SOLD TS ON TORDER.ITEMID=TS.ITEMID

--8

--WHICH ARE THE MERCHANDISE ITEMS WITH A LIST PRICE OF MORE THAN

--$50 AND NO SALES IN JULY

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 SI.SALEID=S.SALEID

WHERE M.LISTPRICE >50

AND M.ITEMID NOT IN (SELECT SI.ITEMID

FROM PET..SALE S INNER JOIN PET..SALEITEM SI

ON S.SALEID=SI.SALEID

WHERE 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

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

FROM PET..MERCHANDISE M RIGHT OUTER JOIN PET..SALEITEM SI ON M.ITEMID=SI.ITEMID

RIGHT OUTER JOIN PET..SALE S ON SI.SALEID=S.SALEID

WHERE M.QUANTITYONHAND > 100

AND M.ITEMID NOT IN (SELECT M.ITEMID

FROM PET..MERCHANDISE M RIGHT OUTER JOIN PET..ORDERITEM OI

ON M.ITEMID=OI.ITEMID RIGHT OUTER JOIN PET..MERCHANDISEORDER MO

ON OI.PONUMBER=MO.PONUMBER

WHERE YEAR(MO.ORDERDATE) = 2004)

GROUP BY M.ITEMID, M.DESCRIPTION, M.QUANTITYONHAND

--10

--USE SUBQUERY FOR 9

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

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

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

WHERE M.QUANTITYONHAND > 100

AND M.ITEMID NOT IN (SELECT M.ITEMID

FROM PET..MERCHANDISE M INNER JOIN PET..ORDERITEM OI

ON M.ITEMID=OI.ITEMID INNER JOIN PET..MERCHANDISEORDER MO

ON OI.PONUMBER=MO.PONUMBER

WHERE YEAR(MO.ORDERDATE) = 2004)

GROUP BY M.ITEMID, M.DESCRIPTION, M.QUANTITYONHAND

--11

--SAVE A QUERY TO ANSWER EXCERCISE 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 FIST QUERY AND

--DISPLAYS THE PROPER LABEL.

--CREATES TABLE

CREATE TABLE CATEGORY

(

CATEGORY VARCHAR(50) NOT NULL,

LOW INT NOT NULL,

HIGH INT NOT NULL

)

INSERT INTO CATEGORY

VALUES ('WEAK', 0, 200)

INSERT INTO CATEGORY

VALUES ('GOOD', 200, 800)

INSERT INTO CATEGORY

VALUES ('BEST', 800, 1000)

SELECT \*

FROM CATEGORY

--QUERY FROM EXCERCISE 5

CREATE VIEW TOTAL\_MERCH\_ANIMAL AS

SELECT C.CUSTOMERID, C.LASTNAME, C.FIRSTNAME,

SUM(CA.ANIMALTOTAL + CM.MERCHTOTAL) AS GRANDTOTAL

FROM PET..CUSTOMER C INNER JOIN CUS\_MERCH\_TOTAL CM ON C.CUSTOMERID = CM.CUSTOMERID INNER JOIN

CUS\_ANIMAL\_TOTAL CA ON CM.CUSTOMERID = CA.CUSTOMERID

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

--RETRIEVES EXCRCISE 5 TABLE DATA ONTO NEW TABLE WITH CATEGORY

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

FROM CATEGORY C INNER JOIN TOTAL\_MERCH\_ANIMAL TMA ON TMA.GRANDTOTAL > C.LOW AND TMA.GRANDTOTAL <= C.HIGH

ORDER BY TMA.CUSTOMERID ASC

--12

--LIST ALL SUPPLIERS (ANIMALS AND MERCHANDISE) WHO SOLD US ITEMS IN JUNE

--IDENTIFY WHETHER THEY SOLD US ANIMALS OR MERCHANDISE

--LISTS MERCHANDISE AND SUPPLIER

SELECT S.SUPPLIERID, S.NAME, M.DESCRIPTION AS ORDERTYPE

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

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

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

WHERE MONTH(MO.ORDERDATE) = 6

UNION

--LISTS ANIMALS AND SUPPLIER

SELECT S.SUPPLIERID, S.NAME, A.CATEGORY AS ORDERTYPE

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

INNER JOIN PET..ANIMALORDERITEM AOI ON AO.ORDERID=AOI.ORDERID

INNER JOIN PET..ANIMAL A ON AOI.ANIMALID=A.ANIMALID

WHERE MONTH(AO.ORDERDATE)=6

--13

DROP

--14

--WRITE A QUERY TO INSERT THE FIRST ROW OF DATA FOR THE TABLE IN EXCERCISE 11

INSERT INTO CATEGORY

VALUES(1,'WALKINS','WALKINS',$2261.51,'BEST')

--15

--WRITE A QUERY TO CHANGE THE HIGH VALUE TO 400 IN THE FIRST ROW OF THE TABLE IN EXCERCISE 11

UPDATE CATEGORY

SET HIGH = 400

WHERE HIGH =800

--17

--CREATE A QUERY TO DELETE THE FIRST ROW OF THE TABLE IN EXCERCISE 11

DELETE CATEGORY

WHERE CUSTOMERID=1

--18

--CREATE COPY OF EMPLOYEE TABLE STRUCTURE

--USE DELETE QUERY TO REMOVE ALL DATA FROM THE COPY

--WRITE A QUERY TO COPY FROM THE ORIGINAL EMPLOYEE TABE INTO THE NEW ONE

CREATE TABLE EMPLOYEE\_CATEGORY

(

CUSTOMERID INT NOT NULL,

LASTNAME VARCHAR(50) NOT NULL,

FIRSTNAME VARCHAR(50) NOT NULL,

GRANDTOTAL INT NOT NULL,

CATEGORY VARCHAR (50)

)

INSERT INTO EMPLOYEE\_CATEGORY

VALUES(1,'WALKIN','WALKIN',$2261.51,'BEST')

INSERT INTO EMPLOYEE\_CATEGORY

VALUES(2,'CUMMINGS','BRENT',$393.12,'GOOD')

INSERT INTO EMPLOYEE\_CATEGORY

VALUES(3,'LOGAN','DWIGHT',$401.18,'GOOD')

SELECT\*

FROM EMPLOYEE\_CATEGORY

--DELETES ALL ROWS

DELETE FROM EMPLOYEE\_CATEGORY