# A8

CIS 310-01

NICOLE SMITH

```
--1. List the products with a list price greater than the average list price of all products.
SELECT ItemID, Description, ListPrice
FROM   PET..Merchandise
WHERE ListPrice > (SELECT AVG(ListPrice) FROM PET..Merchandise)

--2. Which merchandise items have an average sale price more than 50 percent higher than their
average purchase cost?
SELECT M.ItemID, AVG(OI.Cost) AS AvgCost, AVG(SI.SalePrice) AS AvgSalePrice
FROM   (PET..OrderItem OI INNER JOIN PET..Merchandise M ON OI.ItemID = M.ItemID)
                INNER JOIN PET..SaleItem SI ON M.ItemID = SI.ItemID
GROUP BY M.ItemID
HAVING          AVG(SI.SalePrice) > 1.5*AVG(OI.Cost)

--3. List the employees and their total merchandise sales expressed as a percentage of total
merchandise sales for all employees.
SELECT S.EmployeeID, E.LastName, SUM(SI.SalePrice*SI.Quantity) AS TotalSales,
(SUM(SI.SalePrice*SI.Quantity)/(SELECT SUM(SALEPRICE*QUANTITY) FROM PET..SaleItem))*100
AS PctSales
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 S.EmployeeID, E.FirstName, E.LastName

--4. On average, which supplier charges the highest shipping cost as a percent of the
merchandise order total?
CREATE VIEW POCost AS
SELECT PONumber, SUM(Quantity*Cost) AS 'PONumTotal'
FROM   Pet..OrderItem
GROUP BY        PONumber

CREATE VIEW AvgShippingCost AS
SELECT SupplierID, AVG(MO.ShippingCost/PCT.PONumTotal)*100 AS 'PctShipCost'
FROM   Pet..MerchandiseOrder MO INNER JOIN POCost PCT ON MO.PONumber = PCT.PONumber
GROUP BY SupplierID

SELECT SC.SupplierID, S.Name, PctShipCost
FROM   AvgShippingCost SC INNER JOIN PET..Supplier S ON SC.SupplierID = S.SupplierID
WHERE PctShipCost = (SELECT MAX(PctShipCost) FROM AvgShippingCost)

--5. Which customer has given us the most total money for animals and merchandise?
CREATE VIEW AnimalPurch AS
SELECT CustomerID, SUM(SalePrice) AS 'AnimalTotal'
FROM   PET..Sale S INNER JOIN PET..SaleAnimal SA ON S.SaleID = SA.SaleID
GROUP BY CustomerID

CREATE VIEW MerchPurch AS
SELECT CustomerID, SUM(SalePrice*Quantity) AS 'MerchTotal'
FROM   PET..Sale S INNER JOIN PET..SaleItem SI ON S.SaleID = SI.SaleID
GROUP BY CustomerID

SELECT TOP 1 C.CustomerID, C.LastName, C.FirstName, AP.AnimalTotal, MP.MerchTotal,
SUM(AP.AnimalTotal + MP.MerchTotal) AS GrandTotal
FROM   PET..Customer C INNER JOIN MerchPurch MP ON MP.CustomerID = C.CustomerID INNER
JOIN AnimalPurch AP ON AP.CustomerID = C.CustomerID
GROUP BY C.CustomerID, C.LastName, C.FirstName, AP.AnimalTotal, MP.MerchTotal
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?
```

```
CREATE VIEW CustPurchMAY AS
SELECTS.CustomerID, SUM(SI.SalePrice * SI.Quantity) AS 'MayPurch'
FROM   PET..SaleItem SI INNER JOIN PET..Sale S on SI.SaleID = S.SaleID
WHEREMONTH(S.SaleDate) = 5
GROUP BY S.CustomerID

CREATE VIEW CustPurchOCT AS
SELECTS.CustomerID, SUM(SI.SalePrice * SI.Quantity) AS 'OctPurch'
FROM   PET..SaleItem SI INNER JOIN PET..Sale S on SI.SaleID = S.SaleID
WHEREMONTH(S.SaleDate) = 10
GROUP BY S.CustomerID

SELECTCPO.CustomerID, C.FirstName, C.LastName, CPM.MayPurch AS 'MayTotal'
FROM   CustPurchMAY CPM INNER JOIN CustPurchOCT CPO ON CPM.CustomerID =
CPO.CustomerID
                INNER JOIN PET..Customer C ON C.CustomerID = CPO.CustomerID
WHERECPM.MayPurch > 100 AND CPO.OctPurch > 50

--7. What was the net change in quantity on hand for premium canned dog food between January
1 and July 1?
CREATE VIEW PurchasedItems AS
SELECTM.Description, OI.ItemID, Sum(OI.Quantity) AS Purchased
FROM   PET..MerchandiseOrder MO INNER JOIN PET..OrderItem OI ON MO.PONumber =
OI.PONumber INNER JOIN PET..Merchandise M ON M.ItemID = OI.ItemID
WHEREMO.OrderDate BETWEEN '01-01-2004' AND '07-01-2004'
GROUP BY M.Description, OI.ItemID
HAVING M.Description = 'Dog Food-Can-Premium'

CREATE VIEW SoldItems AS
SELECTM.Description, M.ItemID, Sum(SI.Quantity) AS Sold
FROM   PET..Merchandise M INNER JOIN PET..SaleItem SI ON M.ItemID = SI.ItemID INNER JOIN
PET..Sale S ON S.SaleID = SI.SaleID
WHERES.SaleDate BETWEEN '01-01-2004' AND '07-01-2004'
GROUP BY M.Description, M.ItemID
HAVING          M.Description = 'Dog Food-Can-Premium'

SELECTPI.Description, PI.ItemID, PI.Purchased, SI.Sold, Purchased-Sold AS NetIncrease
FROM   PurchasedItems PI INNER JOIN SoldItems SI ON PI.ItemID = SI.ItemID

--8. Which are the merchandise items with a list price of more than $50 and no sales in July?
SELECTM.ItemID, M.Description, M.ListPrice
FROM   PET..Merchandise M
WHEREM.ListPrice > 50 AND M.ItemID NOT IN (SELECT M.ItemID 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 MONTH(S.SaleDate) = 7 )
ORDER BY M.ItemID DESC

--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 DISTINCT M.ItemID, M.Description, M.QuantityOnHand
FROM   PET..Merchandise M LEFT OUTER JOIN PET..OrderItem OI ON M.ItemID = OI.ItemID
                LEFT OUTER JOIN PET..MerchandiseOrder MO ON OI.PONumber = MO.PONumber
WHEREM.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 ItemID NOT IN (SELECT OI.ItemID
                                                FROM
        PET..MerchandiseOrder MO INNER JOIN PET..OrderItem OI ON MO.PONumber =
OI.PONumber

WHERE   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,
LOW INT NOT NULL,
HIGH INT NOT NULL,
PRIMARY KEY (CATEGORY)
)

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

SELECT C.CustomerID, C.LastName, C.FirstName, GTP.GrandTotal, CATEGORY
FROM   GTPurch GTP INNER JOIN PET..Customer C ON GTP.CustomerID = C.CustomerID,
CATEGORY
WHERE GTP.GrandTotal BETWEEN LOW AND HIGH

--12. List all suppliers (animals and merchandise) who sold us items in June. Identify whether they
sold use animals or merchandise.
SELECT S.Name, 'ANIMAL' AS OrderType
FROM   PET..Supplier S INNER JOIN PET..AnimalOrder AO ON S.SupplierID = AO.SupplierID
WHERE MONTH(AO.OrderDate) = 6
UNION ALL
SELECT S.Name, 'MERCHANDISE' AS OrderType
FROM   PET..Supplier S INNER JOIN PET..MerchandiseOrder MO ON S.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,
LOW INT NOT NULL,
HIGH INT NOT NULL,
PRIMARY KEY (CATEGORY)
)

--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 HIGH = 200

--17. Create a query to delete the first row of the table in Exercise 11.
```

```
DELETE FROM CATEGORY
WHERE CATEGORY = 'WEAK'

--18. 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    CATEGORY
FROM   PET..Employee

DELETE FROM CATEGORY

INSERT INTO CATEGORY
SELECT*
FROM   PET..Employee

SELECT * FROM CATEGORY
```