9 Murach’s SQL Server 2016 for Developers: My Guitar Shop exercises

Chapter 6

How to code subqueries

Exercises

1.Write a SELECT statement that returns the same result set as this SELECT statement,but don’t use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

SELECT Phone

FROM Addresses a JOIN Customers c ON c.CustomerID = a.CustomerID

WHERE c.LastName > 's'

ORDER BY c.CustomerID

2.Write a SELECT statement using a subquery to answers the following question: Which ordered items have a DiscountAmount that’s smaller than the average DiscountAmount?

Return the ItemID and DiscountAmount columns for each ordered item.

Sort the results by the DiscountAmount column in descending sequence.

3.Write a SELECT statement that returns three columns: EmailAddress, FirstName, and LastName from theCustomers table.

Return one row for each customer that has not made an order after 2016-04-01 in the Orders table. To do that, use a subquery introduced with the NOT EXISTS operator.

4.Write a SELECT statement that returns three columns: State, City, and the order total for each State and City combination. To do this, you can group the result set by the State and City columns. In addition, you must calculate the order total from the columns in the OrderItems table.

Write a second SELECT statement that uses the first SELECT statement in its FROM clause. The main query should return three columns: State, City and the average order for each city. To do this, you can group the result set by the State and City columns. Finally, order the result by State and then City in ascending sequence. Hint: Join relevant tables in the first SELECT statement.

5.Write a SELECT statement that returns five columns: OrderID, Line1, Line2, City, State, and ZipCode for each order that has a unique shipping address. In other words, don’t include ordersthat have the same shipping addresses as another order.

Sort the results by the ZipCode column.

6.Use a correlated subquery to return one row per customer, representing the customer’s most current order (the one with the latest date) with amount of sales tax greater than 100 dollars. Each row should include these four columns:EmailAddress, OrderID, OrderDate, and TaxAmount.

Answers:

SELECT a.Phone

FROM Addresses AS a

WHERE a.CustomerID IN

(SELECT c.CustomerID FROM Customers AS c

WHERE a.CustomerID = c.CustomerID AND c.LastName>'s')

SELECT i.ItemID, i.DiscountAmount

FROM OrderItems AS i

WHERE i.DiscountAmount < (SELECT AVG(c.DiscountAmount) FROM OrderItems AS c)

/\*-----------Not exists-------------------------\*/

SELECT c.EmailAddress, c.FirstName, c.LastName, c.CustomerID, o.OrderDate

FROM Customers AS c, Orders AS o

WHERE c.CustomerID = o.CustomerID AND

NOT EXISTS

(SELECT o.OrderDate FROM Orders AS o

WHERE o.CustomerID = c.CustomerID AND o.OrderDate >'2016-04-01'

)

/\*------------------Type 1 A-----------------------\*/

SELECT c.EmailAddress, c.FirstName, c.LastName, c.CustomerID

FROM Customers AS c

WHERE NOT EXISTS

(SELECT o.OrderDate FROM Orders AS o

WHERE o.CustomerID = c.CustomerID AND o.OrderDate >'2016-04-01'

)

/\*-------------------Type 1 B---------------------\*/

/\*--------------------------------------------------------\*/

/\*-----------------order total------------------\*/

SELECT a.State, a.City, COUNT(oi.OrderID) AS OrderTotal

FROM Addresses AS a, OrderItems AS oi

GROUP BY a.State, a.City, oi.OrderID

/\*------------------Part B order total--------------------\*/

SELECT a.State, a.City, COUNT(oi.OrderID) AS OrderTotal

FROM Orders AS o

JOIN Addresses AS a ON a.CustomerID = o.CustomerID

JOIN OrderItems AS oi ON oi.OrderID = o.OrderID

GROUP BY a.State, a.City, oi.OrderID

/\*----------Part C avg order-------------\*/

SELECT

a.State,

a.City,

AVG(o.OrderID) avgOrder

FROM Addresses a

INNER JOIN Orders o ON o.CustomerID = a.CustomerID

GROUP BY a.State, a.City

ORDER BY a.State, a.City

/\*---------------Uniq values----------------\*/

SELECT DISTINCT o.ShipAddressID, a.Line1, a.State, a.ZipCode

FROM Orders AS o, Addresses AS a

WHERE o.CustomerID = a.CustomerID

ORDER BY a.ZipCode

/\*------------- Correlated sub query---------------------\*/

SELECT c.EmailAddress, o.OrderID, o.OrderDate, o.TaxAmount

FROM Customers c INNER JOIN Orders o on o.CustomerID = c.CustomerID

WHERE o.OrderDate = (SELECT MAX(o2.OrderDate)

FROM Orders o2

WHERE o2.CustomerID = o.CustomerID

AND o2.TaxAmount > 100)